UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITECNICA SUPERIOR

Ingeniería Técnica de Telecomunicación:
Sistemas de Telecomunicación

Proyecto Fin de Carrera

Desarrollo de un tesauro sobre deficiencia auditiva a partir de patrones en lenguaje natural

Alumno: Pablo Jiménez Arroyo
Tutor: Jorge Luis Morato Lara
Codirector: Valentín Moreno Pelayo

Enero 2016

Título: Desarrollo de un tesauro sobre deficiencia auditiva a partir de patrones en lenguaje natural

Autor: Pablo Jiménez Arroyo

Tutor: Jorge Luís Morato Lara
Codirector: Valentín Moreno Pelayo

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Trabajo Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

_____.

Fdo: Presidente Fdo: Vocal Fdo: Secretario

Agradecimientos

A mi padre...

A mi madre...

A mi hermana...

A mis amigos...

A mis profesores...

A mis tutores...

Al tribunal...

A la Universidad Carlos III...

...y a la vida.

Resumen

Este trabajo pretende aportar un pequeño grano de arena al proceso de dotar de significado semántico a la cantidad de información que cada día se mueve por la web.

La ingente cantidad de bits que se almacenan en los servidores de todo el mundo hace necesario que se les dote de un significado a partir del cual la comunicación entre máquinas y humanos sea más intuitiva y amigable. Más allá de esto, sería interesante que la propia comunicación entre máquinas ponga en juego el contenido semántico que hay detrás de aquello que comparten.

El camino hacia la llamada Web Semántica está aún por recorrer pero el proceso ya se ha iniciado. Tal vez algo de lo aquí expuesto sirva para dar un pequeño paso más.

Abstract

This paper provides a small input to the process provide semantic meaning to the information each day moves through the web.

The huge amount of bits that are stored on servers around the world makes it necessary to equip them with a meaning from which the communication between humans and machines more intuitive and friendly. Beyond this, it would be interesting that the very communication between machines put into play the semantic content behind what they share.

The road to the Semantic Web is still to go but the process has already begun. Perhaps some of the above analysis serves to give a small step.

0.INDICES

Índice de Contenido

Índice de Tablas de resultados

1.INTRODUCCIÓN

Desde hace aproximadamente 25 años, momento en que se comenzó a desarrollar lo que hoy en día conocemos como internet gracias a las ideas y dedicación de Tim Berners-Lee y su equipo, la evolución de esta, de sus contenidos y, sobre todo, de la forma de interactuar con ella que tenemos los humanos ha ido cambiado considerablemente.

Nadie concibe ya que la web sea solo una versión moderna de nuestras antiguas enciclopedias, aquellas con más de veinte tomos que adornaban las estanterías de las casas para ser consultadas en la elaboración de algún trabajo académico o incluso para saber cómo se escribía una palabra.

Más allá de ser un contenedor de información, las posibilidades de la web se han multiplicado en nuestro tiempo y a ella acudimos cada día para relacionarnos socialmente, consumir contenidos multimedia, comprar entradas para conciertos o realizar movimientos desde nuestras cuentas bancarias.

Cada día se hace más necesario que las máquinas sean capaces de tratar los conceptos que hay detrás de la enorme cantidad de datos que manejan. Es preciso que los sistemas puedan interpretar lo que el humano que interacciona con ellos quiere decir. Es más, se hace necesario que las propias máquinas puedan hablar entre ellas no solo tratando largas cadenas de bits sino entendiendo el significado semántico que estas esconden.

El lenguaje, en sus diferentes idiomas y formas que pueden ir más allá de la escrita o hablada, está lleno de conceptos que de alguna manera deben ser clasificados y organizados para que se pueda empezar a trabajar en el desarrollo de una web que los incluya y sea capaz de manejarlos, bien para comunicarse con un humano o bien para interactuar a nivel conceptual con otra máquina.

Las ontologías y los tesauros como formas de estructurar vocabularios más o menos concretos, según el caso, y de establecer relaciones entre los elementos que los forman, son herramientas adecuadas para ayudar en esta labor.

2. OBJETIVOS

2.1 OBJETO DEL TRABAJO

Analizar un corpus especializado de textos en habla inglesa sobre el campo de conocimiento de la Deficiencia Auditiva Genética para poder presentar una aproximación a un tesauro especializado sobre el tema. Los resultados obtenidos podrán ser validados por un experto en el área en un futuro.

2.2 OBJETIVOS

Con el fin de alcanzar el objeto propuesto en el punto anterior se establecen los siguientes objetivos:

- estudiar la capacidad de una larga lista de cadenas de palabras (que a partir de este momento llamaremos "cadenas intermedias") en lengua inglesa para establecer ocho tipos de relaciones entre sustantivos. Las relaciones estudiadas serán las que nos dicen si dos sustantivos son *hiperónimos*, *hipónimos*, *holónimos*, *merónimos*, *antónimos* o *derivados* y dos relaciones que de momento llamaremos "tener como *miembro a*" y "pertenecer a un *dominio*". Para cada tipo de relación se tratarán 999 cadenas intermedias distintas que serán buscadas en una amplia colección de más de 750 artículos en formato PDF que se nos han proporcionado para el estudio y que tratan el tema de la enfermedad auditiva genética. A partir de las cadenas encontradas se mirará qué par de posibles palabras estaría relacionando y se cotejará dicha posible relación consultando la base de datos de Wordnet (base de datos léxica del idioma inglés) que nos fue también proporcionada al comienzo de este proyecto.

- como resultado en parte del proceso anterior, se creará una base de datos de palabras que no se encuentran dentro de las recogidas en la base de datos de Wordnet y que serían la base para la creación de un posible tesauro especializado sobre el campo de conocimiento de la enfermedad auditiva genética. Se intentará establecer además un grado de fiabilidad de que dichos pares de palabras establezcan entre ellas una relación de las ocho estudiadas en el punto anterior.

Los resultados de ambos puntos quedarán almacenados en una base datos que se creará para tal fin.

2.3 ESTRUCTURA DE LA MEMORIA

El contenido de esta memoria se ha divido en diferentes partes que son detalladas en este punto.

Tras el índice y una breve introducción presentamos en el capítulo 2 los objetivos de este proyecto, su estructura, planificación y estimación de costes asociados al mismo. Después de esto llegamos al capítulo 3 denominado “Estado del arte”. En este capítulo se exponen algunos de los conceptos que se manejan en este proyecto. Conceptos tales como qué es una Ontología, un tesauro, a qué se denomina Web Semántica y qué tipos de relaciones semánticas podemos encontrar entre palabras. También se habla un poco sobre Wordnet y sus contenidos.

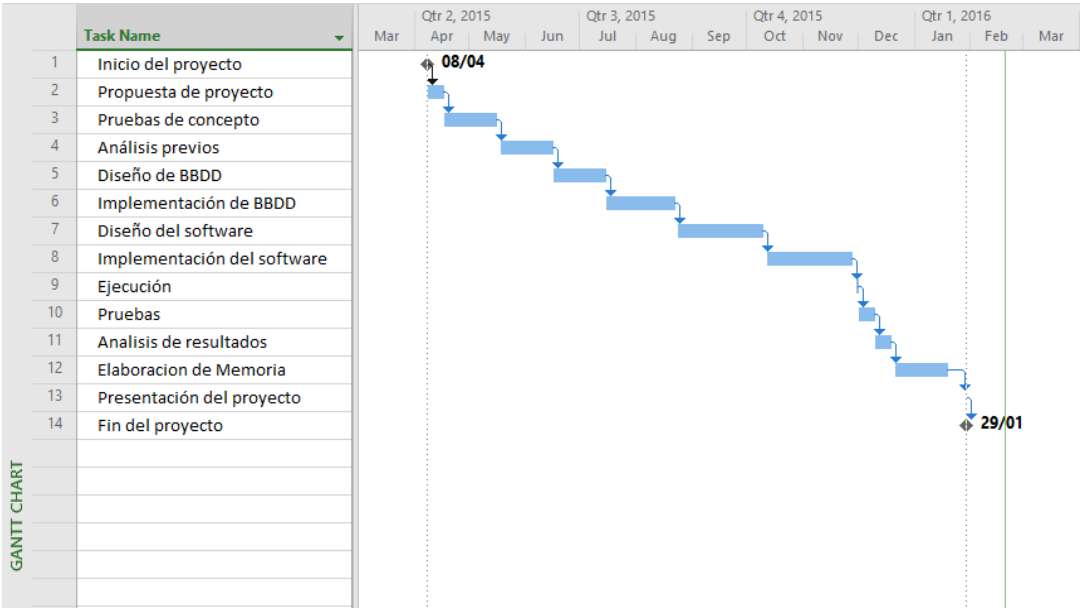
En el capítulo 4, “Herramientas”, se indican qué herramientas y tecnologías se han usado para trabajar estos meses durante la elaboración del proyecto. Se hace referencia a ellas y se da un explicación de porqué han sido seleccionadas entre otras candidatas.

En el capítulo 5, “Diseño”, se presenta el proceso que se ha ido siguiendo desde los datos iniciales de entrada con los que se contaba hasta la obtención de los resultados finales. Se habla del software desarrollado para este proyecto y de sus diferentes partes y funciones.

El capítulo 6 llamado “Resultados” muestra los resultados obtenidos. Por último, la conclusiones se presentan en el capítulo 7 y la bibliografía consultada en el capítulo 8.

2.4 PLANIFICACION

La planificación estimada para la elaboración de este proyecto es la que se puede ver en el siguiente diagrama de Gantt:



2.5 ESTIMACION DE COSTES

La estimación de costes para este proyecto la podemos dividir en:

- Costes en personal
- Costes en hardware y licencias
- Costes en material fungible

Costes en personal

En este proyecto ha trabajado únicamente el propio autor, desempeñando todos los roles que han sido necesario en cada fase del proceso: análisis, diseño y programación.

El salario estimado de un ingeniero senior se ha estimado en unos 30 euros por hora. Atendiendo a la planificación del proyecto vemos que se han empleado 205 días en su realización. Al compatibilizar su elaboración con un trabajo a jornada completa en otra empresa, el número de horas semanales que se le podían dedicar al proyecto, en media, ha sido de 14 horas/semana. Esto da una media de 2 horas/día. Siendo así:

Ingeniero: 205 días X 2 horas = 410 horas.
410 horas X 30 euros/hora = 12300 euros.

Costes en hardware y licencias.

Para la realización de este proyecto se adquirió un equipo nuevo (Intel Pentium i5 - 16GB de RAM) cuyo costé fue de 900 euros. De la cantidad anterior y suponiendo un periodo de amortización de 5 años para el equipo, tenemos un coste imputable al proyecto de aproximadamente 105 euros.

Por otro lado, para la realización de este proyecto se utilizó como S.O. Windows 8.1, cuya licencia costó 85 euros, de los cuales podemos imputar al proyecto un total de 10 euros aproximadamente.

Costes en material fungible

Los gastos en material fungible derivados del uso de diverso material de oficina, tinta de impresora, etc…ascienden a 20 euros.

Presupuesto total del proyecto

Teniendo en cuenta lo anterior, el coste total del proyecto sería:

Recurso	**Coste**
Ingeniero	12300 euros
Equipo y licencias	115 euros
Material fungible	20 euros
Coste Total	12435 euros

3. ESTADO DEL ARTE

3.1 CONCEPTO DE WEB SEMÁNTICA

Si bien en su nacimiento allá por el 1998 la World Wide Web (WWW) se presentó como un conjunto de documentos y enlaces de hipertexto y fue pensada para ser consultada por humanos, poco a poco se ha ido haciendo patente la necesidad de poder procesar la información que contiene de manera automática.

Podemos decir que en su primera versión la web era un almacén de información. Podíamos ir navegando a través de enlaces que nos presentaban dicha información y nos conducían a su vez a otras nuevas a golpe de 'click'.

Tras esto se fue dotando a la web de la posibilidad de que los usuarios aportasen contenidos y ofreciesen servicios. Apareció la capacidad de descargar e intercambiar archivos, de contactar con otras personas a través de redes sociales, de comunicarse en tiempo real a través de videoconferencia y muchos otros servicios que a día de hoy nos son familiares.

Actualmente la Web esta está presente en nuestras vidas para realizar todo tipo de actividades: desde consultar el significado de una palabra en otro idioma, sacar unas entradas para el cine o ponernos en contacto con un amigo que está al otro lado del océano. Esto ha hecho que se haya convertido en un éxito y que cada día presente a sus usuarios millones de recursos según las necesidades de cada uno.

En lo que es ya su tercera versión, la que viene a denominarse web semántica, entra en juego la posibilidad de que ya no sean solo los humanos los que consumen los contenidos de la web sino que las computadoras sean también capaces de procesar su significado y que la relación entre el humano y la máquina sea más amigable. Se busca que la inmensa red de datos que es la Web (semántica) pueda ser procesada de manera directa o indirecta por máquinas. El objetivo principal es abandonar la idea de simple repositorio de documentos para llegar a una Web de conceptos donde sean estos los que estén relacionados entre sí y permitan obtener información de manera más precisa.

Aunque hemos dicho que esta tercera versión de la web podría denominarse "web semántica", no existe realmente unanimidad al respecto. Algunos autores consideran que el concepto de "web semántica" es todavía un proceso en desarrollo y una manera de entender la web mientras que podría decirse que "web 3.0" haría referencia a un cambio tecnológico con respecto a la versión anterior.

Tenemos entonces que la evolución de la web quedaría resumida en tres etapas:

Web 1.0: almacén de información que era consumida por los humanos a través de miles de referencias.

Web 2.0: aparece la posibilidad de que las personas interaccionen entre sí a través de ella, que aporten contenidos, que intercambien archivos, etc.

Web 3.0 o ¿semántica?: dotar de significado semántico al contenido de la web y hacer posible que las máquinas hagan uso del mismo para comunicarse entre ellas.

La información que se encuentra hoy en día en Internet es de una magnitud enorme. Esto ha hecho que el W3C (World Wide Consortium) proponga la introducción de tecnologías que doten de semántica a las páginas Web. De esta forma han surgido el Resource Description Framework (RDF) y el Resource Description Framework Schema (RDF-S).

Resource Description Framework (RDF)

En la actualidad la Web está basada en páginas escritas en HTML (HyperText Markup Language), lenguaje de marcas para implementar hipertexto. Es útil para establecer el aspecto visual del documento e incluir contenido multimedia en el mismo. El HTML, sin embargo, es limitado a la hora de categorizar y filtrar el contenido que presenta.

El RDF es un lenguaje usado para la presentación de información sobre recursos en Internet. Se recoge en seis recomendaciones del W3C: Primer, Concepts, Syntax, Semantics, Vocabulary y Test Cases.

Pretende presentar los objetos como poseedores de propiedades que tendrán a su vez valores. Su mayor utilidad es la presentación de metadatos (fechas, autores, modificaciones de datos en páginas web, licencias, etc…)

El W3C pensó en RDF como herramienta para el procesado e intercambio de metadatos.

Nos va a permitir establecer relaciones semánticas entre distintas URIs dándole a cada una un conjunto de propiedades y valores.

RDF está implementado sobre XML(eXtensible Markup Languaje).

Mientras que en XML tenemos los DTD (Definiciones de Tipo de Documento) y los XMLSCHEMA (esquemas XML), en RDF un esquema aporta información que sirve para interpretar las sentencias del modelo de datos y las restricciones que deberán observarse por estos.

Veamos un típico esquema RDF:

<rdf:RDF>

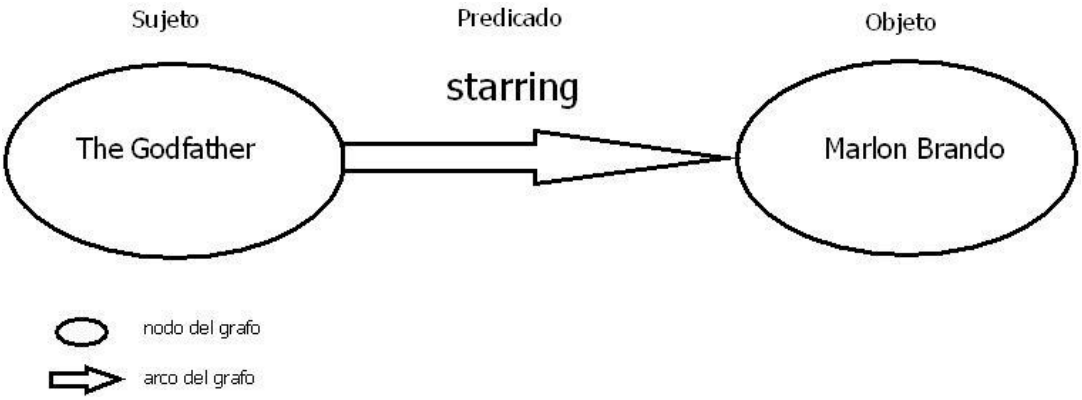
<rdf:Description rdf:about=”http://dbpedia.org/resource/The_Godfather”>

<dbpprop:starring rdf:resource=”http://dbpedia.org/resource/Marlon_Brando”/>

</rdf:Description>

</rdf:RDF>

En RDF, un enlace se compone de tres elementos: sujeto, predicado y objeto.



El atributo “about” en rdf:Description es el sujeto y contiene el nombre de la película. El predicado, dbpprop:starring, muestra lo que significa la relación, en este caso al protagonista de la película. Por último tenemos el objeto, el atributo rdf:resource, que es Marlon Brando. Estos dos conceptos han establecido una relación semántica y están identificados de forma unívoca por URIs (uniform resource identifiers). Es importante entender que el URI <dbpprop:starring rdf:resource=”http://dbpedia.org/resource/Marlon_Brando”/> representa un concepto, no un documento con información sobre el actor. Estas URIs serán desreferenciables, lo cual implica que si la escribimos en el navegador nos llevaría a aquello a lo que hace referencia. En realidad lo que ocurriría sería que una redirección de tipo 303 negociaría el contenido y nos llevaría a la página http://dbpedia.org/page/Marlon_Brando que, ahora sí, contendrá un documento HTML que contendrá datos del actor. Por otro lado, si por ejemplo accediésemos desde una aplicación móvil, nos llevaría a http://dbpedia.org/data/Marlon_Brando que tiene un documento RDF similar al siguiente:

<rdf:RDF>

<rdf:Description rdf:about="http://dbpedia.org/resource/Marlon_Brando">

<rdf:type rdf:resource="http://dbpedia.org/ontology/Person"/>

<rdf:type rdf:resource="http://umbel.org/umbel/rc/Actor"/>

<owl:sameAs rdf:resource="http://es.dbpedia.org/resource/Marlon_Brando"/>

<rdfs:comment xml:lang="en"> Marlon Brando is an American actor.</rdfs:comment>

<foaf:homepage rdf:resource="http://marlonbrando.com/ /"/>

</rdf:RDF>

Tenemos un bloque rdf:Description que se abre para informar sobre un sujeto. Dentro de él podemos incluir varios predicados. En este caso aparecen dos predicados rdf:type que nos ofrecen datos de dos tipos: Marlon Brando es una persona y un actor. Esto, que parece evidente para la inteligencia humana, puede ser de mucha ayuda para una inteligencia artificial. El predicado owl: sameAs nos ofrece otro URI donde se habla de la misma persona pero ahora en un idioma distinto. El predicado rdfs:comment no ofrece enlace alguno, simplemente introduce un texto. Por último tenemos el enlace a la página personal del actor.

Vemos que existen diversos tipos de enlaces RDF: para definir sujetos, predicados, recíprocos, etc…

Usar URI para identificar conceptos en la web semántica es una de las ideas principales en este contexto. Antes hablamos de la redirección 303 y su uso para desreferenciar un URI. De hecho, existen dos formas de desreferenciar los URI: a través de una redirección 303 o a través de un fragmento.

Cuando en el URI exista un fragmento con un '#' este referenciará al concepto. En caso de no llevarlo haría referencia al documento.

Los vocabularios que se utilizan para elaborar RDF se crean con metalenguajes del tipo de SKOS[1], RDFs[2] o OWL[3] con los cuales podemos definir ontologías y tesauros. El uso de dichos metalenguajes hace que no sea necesario crear e implementar APIs para que diferentes webs puedan compartir información sino que esta será compartida de manera sencilla al tener un formato global, común y accesible. No obstante, los vocabularios aportan gran complejidad a la web semántica. Cuando buscamos crear un enlace RDF tenemos que encontrar un lenguaje

[1] SKOS: (Simple Knowledge Organization System)

[2] RDFs: (RDF Schema)

[3] OWL: (Ontology Web Language)

que se adecue al significado de lo que queremos presentar. En caso de no encontrar uno apropiado tendríamos que inventar uno nuevo.

La declaración del espacio de nombres de un vocabulario es un asunto importante. El espacio de nombres aparece antes de los dos puntos en las etiquetas y se usa para que evitar el posible solapamiento al definir dos vocabularios. Tenemos como muestra twitter:image y og:image , que serían idénticos salvo porque van precedidos del espacio de nombres de los vocabularios Twitter Cards y Open Graph . Para declarar estos espacios de nombres usamos un atributo xmlns(XML name space) al comienzo del documento (se pueden omitir si estuviese muy extendido). Un ejemplo sería:

<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#

xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

xmlns:foaf=”http://xmlns.com/foaf/0.1/”>

Aquí se estarían declarando los espacios de nombres rdf, rdfs y foaf.

En un documento HTML podemos publicar un fichero RDF añadiendo en la cabecera la sentencia:

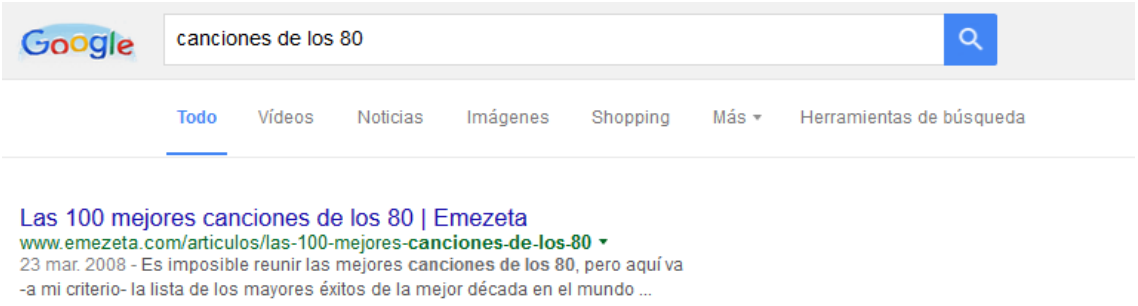
<link rel="alternate" type="application/rdf+xml" href="nobre_fichero.rdf" />

También se podría empotrar dentro del HTML en formato RDFa (RDF con atributos).

Podríamos usar un almacén de datos y lenguaje SPARQL(SPARQL Protocol and RDF Query Language), lenguaje estandarizado para la consulta de grafos RDF y similar a SQL.

En la actualidad Google hace uso de las tecnologías semánticas a la hora de presentar snippets enriquecidos y añadir a su Knowledge graph datos de interés.

Cuando hacemos una búsqueda en Google, esta nos devuelve una sucesión de snippets de la forma: enlace+texto+URL, por ejemplo:



El snippet es el resumen y enlace al sitio web que aparece en la SERP (Search Engine Result Page) o página de resultados de Google. Consta de:

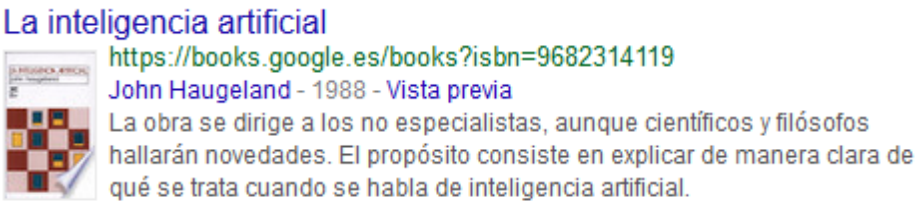
Para crear el snippet, Google obtiene la información de la siguiente manera:

1. De la etiqueta <title> aporta el texto del enlace a la web. Está etiqueta debe haber sido introducida por el desarrollador de la web y es importante que haya sido bien elegida.
2. La etiqueta <description> nos dará las líneas que describen el contenido que tendremos en la web. En caso de no tener dicha etiqueta la página, se incluirá un texto de la misma que contenga las palabras de búsqueda.
3. En enlace en azul es la dirección web que nos llevará a la página a la que hace referencia en snippet.

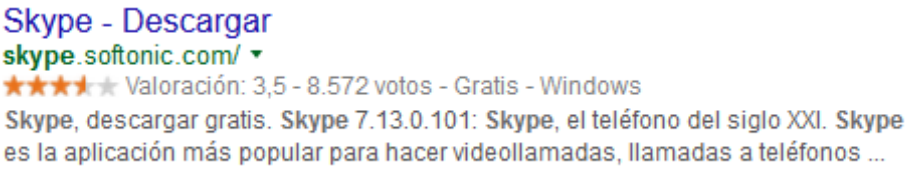
Cuando hablamos de snippets enriquecidos hablamos de resultados como los siguientes al hacer una búsqueda en Google:

Búsqueda de eventos: nos mostraría datos como la fecha, el lugar, la hora de comienzo, etc…

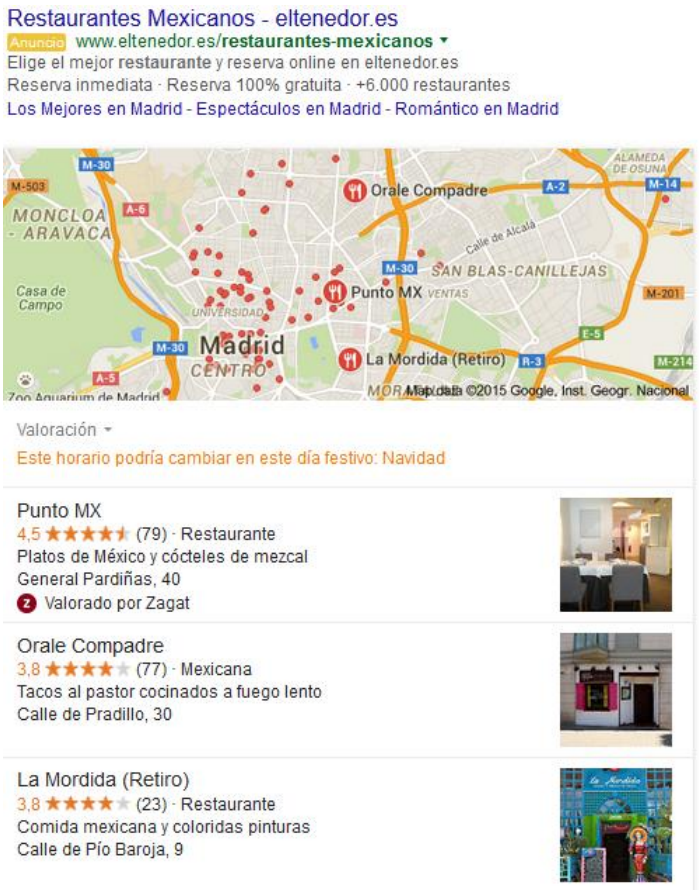
Autores: información sobre el autor de una obra, la foto del documento, años de publicación, etc…



Software: tipo de software, valoración que merece, plataforma para la que está destinada, enlace de descarga, etc…



Opiniones, reseñas, ubicaciones: donde podemos ver opiniones sobre establecimientos, ubicación de los mismos, descripción de sus productos, etc…



Mientras que los metadatos son utilizados para estructurar contenidos, las ontologías y tesauros nos proveen de una semántica para construirlos. Veamos estos dos conceptos.

3.2 ONTOLOGIA

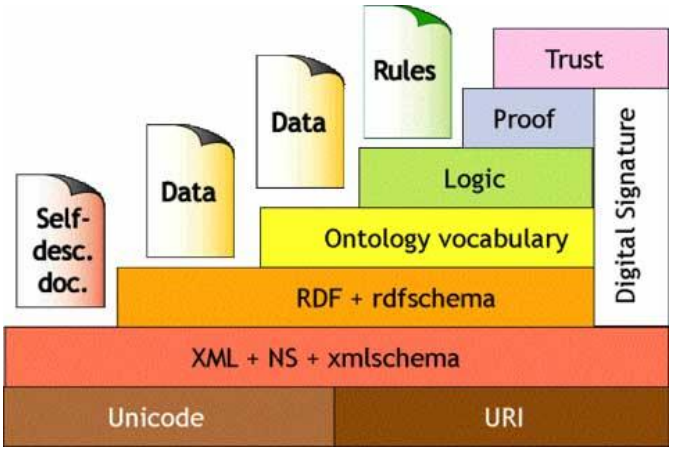
Hablamos de ontología para referirnos a una especificación de una conceptualización. Podemos ver las ontologías como estructuras conceptuales que estarían sistematizadas y consensuadas para poder almacenar, buscar y recuperar la información. Al usar el término conceptualización hacemos referencia a una simplificación del mundo que queremos representar, a una abstracción del mismo. Una ontología no es un programa o base de datos (que tienen sus propios formatos), ni una tabla de contenidos. Una ontología definirá los términos y relaciones básicas que se darán dentro de un área de conocimiento para su compresión y también las reglas para poder combinar dichos términos con el fin de presentar extensiones de estos vocabularios controlados.

La finalidad de la ontología será convertir en conocimiento la información a través de una estructura formalizada que haga referencia a los datos presentando un esquema sobre algún campo de conocimiento que estará normalizado y sobre el que se ha llegado a cierto consenso.

Haciendo uso de metadatos se podrá especificar el esquema de datos que aparecerá en cada instancia y también la forma de realizar deducciones sobre dichos datos, esto es, como crear axiomas aplicables a los distintos dominios de los que trate el conocimiento sobre el que trabajamos.

Haciendo uso de lo anterior, los buscadores web podrían recuperar información al compartir idénticos esquemas de anotaciones web y las aplicaciones encontraran la información adecuada pudiendo crear inferencias de una forma automatizada extrayendo información que tiene relación con la que se encuentra en la web y siguiendo las consignas de las consultas establecidas por los usuarios. Los desarrolladores de servicios y páginas web podrán además realizar un intercambio de datos haciendo uso de estos esquemas y también reutilizarlos.

La ontología, dentro de la pila de capas de la web semántica, estaría situada encima de RDF y de los esquemas RDF:



Algunos de los beneficios a la hora de usar ontologías son:

- Establecen un vocabulario común para representar y compartir conocimiento
- Permiten el intercambio de conocimiento usando un formato.
- Permiten reutilizar conocimiento.
- Dotan de un protocolo específico de comunicación-
- Facilitan la comunicación entre personas.
- Simplifican la traducción de distintas representaciones.

Dentro del contexto de la web semántica, la ontología impulsa el llevar a la red a un espacio de conocimiento y no solo de información. En otros campos, como el de la filosofía, se venía considerando a la ontología como una rama que se encargaba de la organización de la realidad y de la naturaleza de las cosas. A su vez, en el campo de la ingeniería, la inteligencia artificial o la lingüística computacional también es usada como modelo de representación del conocimiento.

Otras definiciones de ontología aportadas por diferentes autores son:

- Una ontología es un vocabulario acerca de un dominio: términos + relaciones + reglas de combinación para extender el vocabulario". Neches, 1991
- "Una ontología es la especificación de una conceptualización". Gruber, 1993. (Aquí el término conceptualización se refiere a un modelo conceptual).
- "Una ontología es una especificación formal de una conceptualización compartida". Borst, 1997. (Aquí el término forma se refiere a que es procesable por ordenador).
- "Una ontología es una base de datos que describe los conceptos generales o sobre un dominio, algunas de sus propiedades y cómo los conceptos se relacionan unos con otros". Weingand, 1997
- Una ontología necesariamente incluirá un vocabulario de términos y una especificación de su significado (definiciones e interrelaciones entre conceptos) que impone estructura al dominio y restringe las posibles interpretaciones. Uschold-Jasper.

Tesauros y ontologías son usadas para organizar distintos campos del conocimiento a través de lenguajes controlados. La principal diferencia entre ambas es que la ontología muestra un mayor nivel de profundización semántica la vez que proporciona una descripción lógica y formal interpretable por máquinas y personas, mientras que el tesauro solo es interpretado por humanos. La ontología permitirá la interoperabilidad entre distintos sistemas.

Tenemos entonces que una ontología será una forma de representar el conocimiento de un ámbito dado sobre el que se aplica una metodología que permite llegar a una representación formal de los conceptos manejados en el mismo y las relaciones entre ellos. Se construye a partir de un contexto de utilización y especificará una forma de ver el mundo o conceptualización. Nos proporcionará, mediante definiciones, vocabulario para hacer referencia a un dominio. Las conceptualizaciones de una ontología (definiciones, categorizaciones, jerarquías, propiedades, herencia, etc.) serán procesables por máquinas.

Antes vimos la definición de Gruber de una ontología. Continuando con este autor [Gruber, 1993.], estarían compuestas por:

- **Conceptos**: serán las ideas más simples y básicas que se van formalizar. Podrán ser métodos, planes, clases de objetos, estrategias, procesos de aprendizaje, etc.
- **Relaciones**: establecen la interacción entre conceptos de un ámbito concreto. Generalmente forman la taxonomía de un dominio. Algunos ejemplos serían: parte de, conectado a, relaciones de hiponimia, meronimia, sinonimia, etc.
- **Funciones**: relación que identifica un elemento mediante el uso de una función que toma en cuenta diferentes elementos de la ontología. Ejemplo: asignar_fecha, categorizar_clase, etc.
- **Reglas** de restricción o axiomas: teoremas declarados sobre las relaciones que habrán de cumplir los distintos elementos dentro de una ontología. Ejemplo: “si X e Y pertenecen a la clase Z, entonces X no es subclase de Y”. Los axiomas hacen posible inferir conocimiento haciendo uso también de la herencia entre conceptos. En la taxonomía de los conceptos dicho conocimiento no estaría presentado de forma explícita. Los axiomas pueden ser de tipo relacional, no-relacional o generales.

Cuando hablamos de ontologías, es importante conocer algunos conceptos importantes relacionados con estas:

- **Clase**: objeto que concreta una categoría y presenta conceptos del ámbito sobre el que se trabaja.
- **Subclase**: aquella que es hija de otra clase. Ella misma es clase al mismo tiempo.
- **Clase jerárquica**: clases que se conectan por la relación "es un tipo de”.
- **Casos** (instancias): casos concretos que pertenecen a una clase. Son los objetos pertenecientes a una clase.
- **Roles o propiedades** : son las propiedades de los conceptos que presentan atributos y características del mismo. Son un complemento que permite especificar mejor las características de las clases.
- **Facetas**: definen el tipo de valor puede contener un rol determinado, su número de valores, cuáles son sus valores válidos, etc. Se les llama también restricciones de roles.

- **Valor**: muestra una propiedad que se asigna a alguna instancia o clase.
- **Tipo**: Presenta el tipo de valor (numérico, booleano, cadena de caracteres, etc.)
- **Cardinalidad**: número de valores posible para un slot individual (tanto el máximo como el mínimo).
- **Herencia**: proceso por el que clases que están definidas en niveles más altos de la jerarquía permiten heredar propiedades y valores a otras subclases e instancias.
- **Variable**: Espacio sin contenido que se completa recurriendo instancias y clases. Las variables comenzarán con el signo de interrogación.
- **Relación**: nuevo conocimiento al que se llega por deducción, partiendo del conocimiento tenemos previamente en la ontología. Las relaciones hacen uso de las variables.

Cuando se diseña una ontología se deben considerar ciertos aspectos clave en su elaboración. Primero la claridad, ya que debe poder comunicar el significado de sus términos de una manera adecuada. Se usarán definiciones que busque el mayor grado de objetividad posible y que puedan explicarse usando un lenguaje natural. Segundo la coherencia, ya que debe hacer posible la realización de inferencias que guarden consistencia en relación a las definiciones. La ontología tendrá que ser también extensible, permitiendo extensiones y especializaciones si nuevos usos lo requieren. Deberá mostrar especificidad, no dependiendo de una codificación concreta a nivel de símbolo. Y por último se buscará la precisión, haciendo la menos cantidad de suposiciones posible del mundo modelado.

Clasificación de las ontologías

Las ontologías se clasifican en varios tipos según distintos aspectos. Algunas de estas clasificaciones son:

- **Atendiendo al ámbito de conocimiento que representan:**
  - **Generales:** situadas en los niveles más altos dado que presentan conceptos más generales (materia, objeto, tiempo, espacio, etc.)
  - **De dominio:** muestran el vocabulario de determinado dominio de conocimiento.
  - **Específicas:** aquellas que están presentan los conceptos de determinado campo y están especializadas.
- **Atendiendo al tipo de agente al que van destinadas:**
  - **Lingüísticas:** vinculadas a características lingüísticas como son las semánticas, sintácticas y gramaticales. Están pensadas para su uso por seres humanos.

- **No lingüísticas:** pensadas para ser usadas por agentes inteligentes y robots.
- **Mixtas:** mezclan características vistas en los puntos anteriores.

- **Atendiendo al nivel de razonamiento lógico y abstracción que permiten:**
  - **Descriptivas:** contienen relaciones entre propiedades y conceptos, descripciones, taxonomías, etc… sin permitir establecer inferencias.
  - **Lógicas:** permiten realizar inferencias lógicas haciendo uso de estrategias como la inclusión de axiomas.

Más allá de estás clasificaciones, algunas de las características que incluyen las ontologías son:

- Podrían combinarse dos o más ontologías (ontologías múltiples) si la finalidad de otra ontología así lo necesita.
- Se puede hablar de topología de ontologías, pudiendo caracterizar una red de ontologías haciendo uso de conceptos como la abstracción o la multiplicidad . Podría implementarse una estrategia que construyese de abajo hacia arriba de forma gradual para ir proporcionando una descripción total del mundo.
- Un concepto se podrá representar de muchas formas, por lo que podrán darse varias representaciones del mismo a la vez. Se habla entonces de multiplicidad de la representación.
- Se puede hacer un mapeo de ontologías, estableciendo relaciones entre elementos de varias de ellas para establecer generalizaciones especializaciones, conexiones, etc.

Usos de las ontologías

Veamos algunos de los diferentes aspectos por los que es interesante el uso de ontologías dentro de distintos campos de la actividad humana:

- Aclaran la estructura del conocimiento: ya que en el proceso de análisis ontológico se definen los conceptos del dominio y sus relaciones, lo que permite una clara especificación de la naturaleza de dichos conceptos y de los términos que se usan para representarlos.
- Reducen la ambigüedad terminológica y conceptual: ya que el análisis ontológico dota de un marco de unificación incluso entre personas con distintas necesidades o puntos de vista que dependen de sus contextos particulares.
- Permiten compartir conocimientos: ya que se consigue un conjunto de conceptualizaciones de un dominio específico y un conjunto de términos que las soportan.

Más concretamente, en la ingeniería del software, se han determinado, por parte de diferentes autores, distintas utilidades:

- Comunicación: reduciendo la ambigüedad conceptual y terminológica ya que proveen un marco de unificación.
- Interoperabilidad: importante dada la existencia de numerosos usuarios y herramientas diferentes dispuestas a intercambiar datos. Las ontologías pueden actuar como herramientas de traducción entre distintos lenguajes y representaciones.
- En ingeniería de sistemas: donde el uso de ontologías puede darse con distintos propósitos como por ejemplo:
    - Especificación: facilitando la identificación de requerimientos y compresión de las relaciones entre componentes. Esto es especialmente importante si existen conjuntos de diseñadores trabajando sobre diferentes dominios.
    - Confiabilidad: las ontologías informales podrían mejorar la confiabilidad de un sistema sirviendo como base para chequear manualmente el diseño contra la especificación. Por otro lado las ontologías formales permitirían el chequeo (semi)automatizado del sistema de software contra la especificación declarativa.
    - Reusabilidad: una ontología eficiente soportará la importación y exportación de módulos entre componentes de software. Caracterizando clases de dominios y tareas dentro de los mismos, las ontologías podrían proveer un marco de trabajo donde se determine qué aspectos de estás pueden ser reutilizados entre dominios y tareas distintas. El objetivo es conseguir librerías de ontologías que se puedan reutilizar y adaptar para distintas clases entornos y problemas.

3.3 TESAURO

Usamos tesauros para poder controlar el vocabulario, para orientar a los indizadores y usuarios sobre términos a usar y para poder lograr una mejor calidad de recuperación de contenido.

Existen tesauros de carácter general y otros de carácter más específico: arte, metalurgia, arquitectura, cocina, etc.

Podríamos decir que un tesauro provee de distintos tipos de información a indizadores y usuarios. La indización es la capacidad de representar el contenido temático de un determinado recurso de información. Esto nos daría como resultado un índice con los términos obtenidos y que se podrán usar para posteriores búsquedas con herramientas de acceso a contenidos y recuperación de información.

Un tesauro contendrá una sería de términos permitidos o *Descriptores* y otros no permitidos o *No descriptores.* Los *Descriptores* serían los términos que se pueden usar en el tesauro, mientras que los *No Descriptores* serían aquellos que son sinónimos de los primeros. A partir del no descriptor se debería mirar qué término sería posible usar en vez de este. Observar los no permitidos a partir de un descriptor puede ayudar a una mejor compresión del significado del término.

Entre los términos descriptores se establecerán una seria de relaciones semánticas que nos ayudaran a dirigirnos entre términos y a llegar al más adecuado.

Para elaborar un tesauro se debe realizar un proceso de recogida de términos. Algunos serán los descriptores a la finalización del proceso y otros, sin serlo, podrían sugerir conceptos que necesitarían ser cubiertos de alguna forma.

Como fuentes de términos podemos encontrar:

- Listas de términos: diccionarios, glosarios, índices, otros tesauros, etc.
- Textos: completos, extractos, especializados, títulos, FAQ, etc.
- Personas: especialistas en temas concretos.

Se busca, en lo posible, sustantivos o sintagmas nominales.

Los términos obtenidos en la recogida de términos, pueden pasar por un proceso de normalización. La siguiente tabla nos da una serie de directrices a utilizar en este punto:

Directrices	**Ejemplos**
Uso del plural para cosas contables	“cajas”
Uso del singular para cosas incontables	“agua”
Singular para procesos, características y condiciones	“congelación” “temperatura” “inestabilidad”
No inversión de términos	“herramientas de bricolaje” (en vez de “ herramientas, bricolaje”)
No abusar de preposiciones	“obras literarias” (y no “obras de literatura”)
Exclusión de puntuación, abreviaturas, caracteres especiales y diacríticos.	“Programas cooperativos” (en vez de co-operativos)

El tesauro tendrá que dejar claro el significado del término. Para las palabras polisémicas (las que tienen más de un significado) se puede especificar de cuál se trata con un calificador: capacidad (eléctrica). También se pude transformar a sintagma nominal: capacidad eléctrica.

En la elaboración de un tesauro, además de los términos recogidos de diversas fuentes, se podrían introducir otros. Tendríamos por ejemplo:

1. Términos que expresen conceptos generales.
2. Términos estructurales.
3. Términos nuevos.

Términos que expresen conceptos generales

Expresan conceptos amplios y son útiles para búsquedas más amplias:

Ejemplo:" Medios de transporte", porque podría usarse para sustituir a "autobuses", "tren", "coche"," avión"…

Términos estructurales

Aquellos que contribuyen a una mejor compresión de la estructura de las relaciones semánticas: "Épocas históricas" para aclarar la relación entre "Época" y "Edad Media".

Términos nuevos

Por ejemplo en la construcción de un tesauro para indizar documentos que no están en formato texto se podrían añadir nuevos términos.

Descriptores y no descriptores

Tras una recogida de términos se debe decidir cuáles son equivalentes. Para la indización y la búsqueda los términos equivalentes serán tratados como si significaran la misma cosa y se representarán por un mismo término descriptor o preferido.

Cuando los términos equivalentes significan la misma cosa se denominan sinónimos. Por otro lado, si los términos equivalentes significan diversas cosas en leguaje ordinario, hablamos de cuasi-sinónimos. Para la recuperación e indización será mejor agruparlos juntos.

Existen diferentes tipos de cuasi-sinónimos:

- cuando hay un solapamiento de significados, estos se tratan como equivalentes: “genios” y “prodigios” podrían considerarse equivalentes.
- cuando un término tienen un alcance que se incluye en el de otro también se pueden tratar como equivalentes. Por ejemplo, “hierro” y “metal”, siempre y cuando no sea necesario distinguir ítems sobre el hierro de ítems sobre otros metales.
- cuando los términos son contrarios a veces también se tratan como equivalentes ya que la consideración de uno es posible que ayude a aclarar el significado de otro: “luz” y “oscuridad”.

Se llamarán descriptores a los términos que contienen toda la información sobre un concepto. Por otro lado, los términos no descriptores serán los que ayuden a definir el alcance de los anteriores.

Un primer tipo de relación dentro de un tesauro sería la relación de equivalencia semántica intralingüística. Con el fin de no tener problemas debidos a la riqueza del lenguaje natural surgieron los lenguajes documentales. La relación de equivalencia semántica intralingüística ayudará a subsanar dichos problemas en los tesauros y se establecerá entre un descriptor con un no descriptor, pertenecientes ambos al mismo idioma.

Este tipo de relación es la que se da entre el descriptor y no descriptor en el momento en que, para realizar la indización, se considera que uno o más términos hacen referencia a un concepto idéntico.

De esta forma se enlaza un descriptor con cada uno de los términos que se utilizarán para referirse al mismo concepto. Los no descriptores estarán dentro del tesauro y servirán para acceder al descriptor con el fin de presentar el concepto. En lo referido a su uso para la indización, se disminuye así el vocabulario del lenguaje documental.

Esta relación de equivalencia intralingüística es recíproca, de tal forma que si el término descriptor A se relaciona con el no descriptor B, B estará relacionado con A.

El término no descriptor se unirá al descriptor por medio de la referencia USE y en dirección opuesta a UP (“utilizado por”). Ejemplo:

ESPECIALIZACIÓN PROFESIONAL	PROFESIONALIZACIÓN
USE PROFESIONALIZACIÓN	UP ESPECIALIZACIÓN PROFESIONAL

Para la elección de descriptores podemos ver unos ejemplos:

Directrices	**Ejemplos**
Uso común	Esperanza de vida UP Longevidad (termino comúnmente usado)
Ambigüedad	APARTAMENTO UP VIVIENDA (apartamento es más preciso que vivienda)
Amplitud	MAMIFEROS UP FELINOS (los felinos son a su vez mamíferos)
Concisión	OKUPA UP MOVIMIENTO OKUPA (una palabra mejor que dos)
Colocación	PRESIDENTE UP POLITICOS (en una lista alfabéticamente ordenada, "presidente" aparecería cerca de "políticos")
Coherencia interna	Si se conviene usar el nombre en latín de ciertas especies, se hará constantemente
Coherencia externa	Uso de un descriptor determinado porque sea el término usado normalmente dentro del ámbito en el que estamos

A veces en vez de un término no descriptor para guiar al usuario o indizador, se puede usar una combinación de términos descriptores. La referencia USE ira en este caso hacía todos los términos descriptores y la UP se marcará de manera especial generalmente.

Ejemplo:

MERCADO MEDIAVAL USE MERCADO + MEDIAVAL
MERCADO UP + MERCADO MEDIAVAL
MEDIAVAL UP +MERCADO MEDIAVAL

A veces se usaran sintagmas (varias palabras) para formar el descriptor. Los casos en los que se justifica este hecho son:

1. Cuando no es posible combinar términos en la fase de búsqueda o indización.
2. Cuando fuesen necesario muchos términos para indizar un concepto o un documento.
3. Cundo el número de términos adecuados no es muy alto.
4. Cuando el sintagma representa mejor en concepto que su partición.
5. Cuando el término es usado frecuentemente en la búsqueda o indización.
6. Cuando los componentes del sintagma aparecen a menudo en diversas relaciones sintácticas: “MUSICA BARROCA”, “MUSICA CONTEMPORANEA”.
7. Cuando el término es necesario en el esquema semántico, en especial cuando otros descriptores representan conceptos más específicos.
8. Cuando hay dudas.

Las relaciones semánticas ayudarán de varias maneras:

1. Posibilitando la generalización o especificación de la búsqueda.
2. Escogiendo un correcto nivel de generalización en la búsqueda e indización.
3. Controlando cuando un término será usado en la indización de un ítem dado o en la formulación de una búsqueda.

Las principales relaciones semánticas utilizadas entre descriptores y no descriptores son las relaciones jerárquicas y no jerárquicas.

Las conexiones TG (término genérico) y las TE (término especifico) se utilizan para indicar relaciones de tipo jerárquico. En este tipo de relaciones, un término se sitúa por encima de otro si es más amplio en alcance. A menudo es útil establecer relaciones jerárquicas en primer lugar al construir un tesauro.

Dentro de las relaciones jerárquicas podemos ver algunos ejemplos:

Relación Genero / Especie o genérica: A será un término genérico de B (siendo B un término específico de A) cuando todas las cosas incluidas en la clase que especifica B se incluyen en la clase nombrada por A. Ejemplo: "FRUTAS" es un término más amplio que "PERAS" (y "PERAS" es un término más específico que "FRUTAS") dado que las peras son un tipo de frutas.

Relación Parte / Todo o partitiva: A es un término genérico de B (y B es específico de A) si todo lo que incluye la clase que nombra B es una parte de algo que incluye la clase nombrada por el término A. Ejemplo: "CUERPO" podría ser un término genérico de "CABEZA" dado que la cabeza forma parte del cuerpo.

Relación enumerativa: se da entre una categoría general de objetos representados por un sustantivo que sea común y un caso de tal categoría que sea individual y que es una clase de un solo objeto caracterizado por un nombre propio.

En un tesauro, las relaciones TG y TE serán generalmente "inversas", es decir, si A es un término más amplio que B, B será más específico que A y viceversa.

Ejemplo: si en un tesauro tenemos la entrada

MARTILLOS

TG HERRAMIENTAS

posiblemente aparezca también

HERRAMIENTAS

TE MARTILLOS

Los tesauros pueden contener términos que tengan varios términos más amplios, esto es, varías referencias TG:

PSICOLOGIA SOCIAL

TG PSICOLOGIA

TG SOCIOLOGIA

A veces podemos llegar a términos que no tengan otros más amplios dentro de un tesauro que cubre determinado tema. Por ejemplo en un tesauro de medicina, "MEDICINA" podría no tener términos más amplios.

Para no sobrecargar un tesauro y hacer difícil su lectura, se pueden omitir conexiones que sean implícitas en otras. Si el término A es más amplio que B y este a su vez es más amplio que C, no se incluirían relaciones TG/TE entre A y Z. Ejemplo:

ANIMALES

TE CARNIVOROS

CARNIVOROS

TE LEONES

no incluyendo

ANIMALES

TE LEONES

La utilidad de las relaciones jerárquicas viene dada por la capacidad de perfeccionar el control del vocabulario que se inicia con la búsqueda de los términos preferidos a través de la inserción de una red de relaciones semánticas que ya tienen significado de por sí.

La relación jerárquica hace más sencilla la navegación vertical en el tesauro. Nos lleva a términos más genéricos o más específicos al realizar una búsqueda y nos permite dotar de la precisión adecuada a esta o al proceso de indización. En la búsqueda podremos acceder al descriptor más específico o bien, si con el mismo no obtuviésemos resultados debido a su especificidad, subir uno o varios niveles hasta encontrar el que buscamos (o bien, al contrario, ir bajando niveles si partimos de un descriptor demasiado genérico que nos devuelve demasiadas referencias y queremos llegar a uno más específico). En la indización podremos también escoger el término más preciso a utilizar.

En un tesauro también podemos encontrar relaciones asociativas TR (término relacionado). Este tipo de relación se da entre términos relacionados conceptualmente sin tener una relación de tipo jerárquico. Cuando sea conveniente recordar a un usuario o indizador que quiera utilizar A que también existe el término B, usaremos una relación TR. Son relaciones que aparecen entre términos que, sin ser equivalentes ni mediar entre ellos relación jerárquica, se asocian mentalmente de tal manera que la conexión entre ellos debe aparecer en el tesauro, de tal manera que a través de ella se podrían obtener otros términos que podrían ser de utilidad en la búsqueda o indización. Es una relación que se da entre descriptores y es recíproca y simétrica. Ejemplo:

PLUMAS

TR CALIGRAFÍA

generalmente también tendrá la entrada:

CALIGRAFÍA

TR PLUMAS

Dentro de los tipos de relación asociativa (TR) podemos diferenciar las relaciones en que ambos términos pertenecen a una misma categoría y aquella en la que pertenecen a categorías diferentes.

Para el primer tipo, la relación asociativa entre términos de una misma categoría, ambos términos estarán relacionados de forma jerárquica con un miembro superior, aunque no habrá relación jerárquica entre ellos. Suele darse entre términos que, sin ser sinónimos, están tan asociados que un usuario que quiera obtener información sobre uno de ellos tendrá en cuenta el otro, aunque cada uno de ellos tiene su definición exacta y no equivalente.

Por otro lado, la relación asociativa de términos que pertenecen a categorías distintas se da entre términos que sería imposible enlazar de otra forma que no sea dentro del tesauro, lo cual hace que sean de gran utilidad. Tiene cierto componente de subjetividad a la hora de establecerse. Un ejemplo de lista de tipos de esta relación sería:

Todo y su parte	Botella y tapón
Causa y efecto	Caída y lesión
Acción y el lugar de la acción	Ejercicio y Gimnasio
Acción y su objeto	Espeleología y cuevas
Acción y su producto	Pastelería y pasteles
Acción y su agente	Categorización y categorizar
Objeto y su propiedad	Venenos y toxicidad
Objeto y aplicación	Termómetros y medición de temperatura
Material y producto	Plata y joyas
Conceptos complementarios	Enseñanza y aprendizaje
Conceptos contrarios	Paciencia e impaciencia

Es interesante apuntar que la relación asociativa permite la navegación horizontal dentro del tesauro entre descriptores que no están asociados jerárquicamente. Muestran relaciones entre términos que están en la mente de los especialistas de cierto campo de conocimiento y que no se pueden enlazar de ninguna otra forma dentro del lenguaje documental. Enriquecen la búsqueda e indización al presentar otros descriptores potencialmente adecuados y útiles.

3.4 WORDNET

Desarrollada a partir de los años 80, Wordnet es una base de datos con información léxico-conceptual de la lengua inglesa. Su contenido está organizado en forma de red semántica en la que aparecen unidades léxicas y las relaciones entre ellas. Su objetivo es convertirse en un modelo léxico-conceptual de conocimiento para los angloparlantes.

La dirección del proyecto corre a cargo de George Miller, psicolingüista de la Universidad de Princeton. Si bien la última versión de Wordnet es la 3.1 (junio-2011), la última versión liberada es la 3.0 (diciembre-2006).

Se puede decir que Wordnet persigue dos objetivos:

1. Validar diferentes teorías psicolingüísticas sobre organización léxica.
2. Posibilidad de ser utilizada en aplicaciones que necesiten acceder a información léxica.

La diferencia más importante entre Wordnet y otros proyectos similares es que es el único que ha tenido como idea principal organizar el léxico en diferentes campos semánticos. De hecho, una de sus motivaciones principales fue poner a prueba distintas teorías lexicológicas y psicolingüísticas relacionadas con la estructura del lexicón mental mediante su implementación directa en un ordenador.

El lexicón mental fue organizado partiendo de un modelo de varias redes semántica desde el cual los investigadores que iniciaron Wordnet se propusieron, en 1985, diseñar una herramienta que hiciese posible desplazarse por un diccionario tanto de forma alfabética como conceptual.

El lexicón estará dividido en cuatro categorías en Wordnet: sustantivos, adjetivos, verbos y adverbios (se podría hablar de una quinta categoría de elementos funcionales). Esta organización hará que Wordnet presente una cantidad considerable de información repetida que no se mostraría en un diccionario común en los casos en que cierta palabra se encontrase en varias categorías.

A su vez, esta estructura hace posible un análisis más fácil de las distintas organizaciones semánticas que se dan entre las categorías sintácticas que hemos indicado.

En Wordnet se denomina *synset* a cada grupo de significado. La versión 3.0 de Wordnet contiene 155.287 palabras agrupadas en 117.659 *synsets* y con 206.941 pares de palabras. El peso de la base de datos es de 12MB.

Wordnet diferencia entre cuatro categorías de palabras: sustantivos, verbos, adjetivos y adverbios ya que tienen distintas reglas gramaticales. En cada *synset* tenemos un conjunto de palabras sinónimas o bien locuciones con un significado concreto. Una palabra/locución podría estar en diferentes *synsets* si expresan significados distintos.

Asociado al *synset* está el concepto de gloss: sucintas definiciones, pequeñas frases o ejemplos que aclaran el significado del *synset*.

Si consideramos el concepto de relación semántica o de significado, encontramos en Wordnet conexiones entre *synset* atendiendo a dicha relación. Tenemos así, para cada categoría de palabras, relaciones de:

Sustantivos: hiperónimos, hipónimos, holónimos y merónimos.

Verbos: hiperónimos, tropónimos, consecuencia lógica y términos coordinados.

Adjetivos: similar a, participios de verbos y sustantivos relacionados.

Adverbios: la raíz de los adjetivos.

Otros datos que podemos encontrar en Wordnet son:

- *Polysemy count*: cuantos *synsets* contienen esa palabra.
- *Frecuency score*: ejemplos con las palabras que aparecen en un synset y un contador que indica con qué frecuencia aparece dicha palabra con un determinado significado.

Como indicamos anteriormente, Wordnet se organiza en base a relaciones de tipo semántico. Los significados vienen representados por los *synsets* y las relaciones entre estos se expresan con punteros (pointers) entre ellos.

Sustantivos

Los sustantivos tienen definiciones que aparecen estructuradas en jerarquías semánticas creadas en base a los términos subordinados que se incluyen en las definiciones de los sustantivos junto a las características propias que distinguen a un sustantivo de su hiperónimo. En Wordnet tenemos unas 57.000 formas nominales con una organización de aproximadamente 48.000 significados. La relación de superordinación crea una estructura de jerarquía semántica que se duplica en el momento que usamos punteros entre synsets. Esta jerarquía no contiene generalmente más de doce niveles de organización ya que está limitada en cuanto a su profundidad. La jerarquía comienza con los niveles inferiores de los términos subordinados y viaja hacia la parte superior donde se encuentran los términos más genéricos. Los rasgos distintivos se introducen creando una estructura de herencia léxica donde el término superordinado permite heredar rasgos distintivos a aquellas palabras de niveles inferiores.

Los sustantivos en Wordnet no se estructuran en una única jerarquía con un término superordinado que sea general y que englobe al resto. Los sustantivos se han agrupado, al contrario que en otras jerarquías, en alrededor de un grupo de conceptos genéricos o "primitivos semánticos donde cada uno es un término de nivel superior de una jerarquía separada. Según sus autores, estas jerarquías representan campos léxicos correctamente definidos y en los que todos ellos tienen su propio vocabulario.

Esta forma de estructurar los sustantivos plantea un problema de status de estos conceptos genéricos. Algunas veces aparecen como elementos léxicos y son tratados de esta forma por

sus autores. Sin embargo en ocasiones son llamados “componentes semánticos primitivos” y se muestran como conceptos a los que se adscriben un campo léxico y los lexemas que aparecen en él. Estos componentes son los necesarios para presentar dominios conceptuales y léxicos, pudiendo significar esta mezcla un problema a la hora de una tratarlos computacionalmente debido a que la línea que separa conceptos y lexemas no queda bien definida.

Otro problema que aparece en estas jerarquías múltiples es decidir qué términos deben ser los conceptos genéricos que aparecerán la parte superior de la jerarquía. Con el fin de abarcar todos los sustantivos de la lengua inglesa, Wordnet ha incluido todos aquellos que se han considerado necesarios.

Los 25 componentes semánticos primitivos que tenemos en Wordnet son:

{act, action, activity}	{feeling, emotion}	{plant, flora}
{animal, fauna}	{food}	{possession}
{artifact}	{group, collection}	{process}
{attribute, property}	{location, place}	{quantity, amount}
{body, corpus}	{motive}	{relation}
{cognition,knowledge}	{natural object}	{shape}
{communication}	{natural phenomenon}	{state, condition}
{event, happening}	{person, human being}	{substance}
		{time}

Estos componentes son independientes pero no exclusivos de forma mutua. Se han formado siendo agrupados a su vez en varias categorías que son más generales a través de relaciones de hiponimia. Por ejemplo, para las cosas intangibles, tenemos que los componentes se pueden agrupar de la siguiente manera:

{plant, flora}
{living thing, organism}
{animal, fauna}
{person, human being}
{thing, entity}

{natural object}

{non-living thing, object}

{artifact}

{substance}

{food}

Adjetivos

Podemos encontrar en Wordnet unas 19.000 formas adjetivales repartidas en aproximadamente 10.000 *synsets*. Los divide en dos categorías principales: descriptivos y relacionales.

Dentro de los adjetivos descriptivos se encontrarían aquellos que dotan al sustantivo de atributos bipolares y que estarían organizados según relaciones de antonimia, o de oposición de significado, y de sinonimia o de similitud de significado. Habrá adjetivos que no tengan antónimos directos pero se considerará que sí los tienen indirectos atendiendo a la similitud de significado con otros adjetivos que sí tengan antónimos directos. Entre los adjetivos que representan un valor de un atributo y el synset de sustantivos que hace referencia a dicho atributo existirán punteros que relacionarán dichos adjetivos y sustantivos.

Cuando tenemos adjetivos que al ser aplicados a un sustantivo expresan algo del tipo "relativo a" o bien "asociado con", serán llamados adjetivos relacionales. Estos adjetivos tendrán asociados punteros referenciando aquellos sustantivos a los que pueden acompañar.

Para los adjetivos de tipo relacional, la antonimia es la relación semántica básica y estará estructurada para que estos adjetivos presenten oposición en los atributos que, como vimos, son bipolares la mayoría de las veces. Esto plantea algunos problemas como por ejemplo el caso de dos adjetivos con significados similares pero con antónimos distintos o adjetivos sin antónimo a los que no se les puede asignar otro adjetivo de significado similar.

Verbos

Para organizar los verbos ingleses no son válidas las relaciones semánticas usadas a la hora de construir las redes de sustantivos y adjetivos. Entre verbos se dan relaciones semánticas que tienen una naturaleza distinta a las observadas en otras categorías.

En Wordnet existen más de 21.000 verbos o formas verbales y unos 8400 "synsets". Se agrupan en 15 grupos distintos atendiendo a ciertas características semánticas. Estos grupos se relacionan con dominios de tipos *cognition*, *consumption* , *communication*, *creation, competition*, *change*, *verbs of bodily care and functions* , *contact*, *emotion*, etc. Los verbos

anteriores hacen referencia a eventos o acciones. También existe un archivo que denota estados. Dentro de esta última categoría tenemos ejemplos como resemble , *suffice* o *belong* . Dichos verbos se refieren a estados, sin compartir ninguna propiedad semántica más y no formando ningún dominio semántico.

Mientras que los sustantivos se organizan siguiendo el principio de herencia léxica y los adjetivos el de oposición bipolar, las relaciones que organizan los verbos en Wordnet usan el principio de implicación léxica. Este principio plantea, siguiendo la lógica proposicional, que si tenemos dos verbos Verbo_1 y Verbo_2, habrá una implicación lógica entre las frases "Alguien Verbo_1" y "Alguien Verbo_2". Por ejemplo, *chew* implicaría léxicamente a *eat* dado que la oración "*He is chewing*" implica "*he is eating*". Es una relación unilateral de implicación léxica, de tal manera que si Verbo_1 implica a Verbo_2, no podrá ser que Verbo_2 implique Verbo_1 salvo que ambos tengan relación de sinonimia.

Mientras que en los sustantivos hablábamos de la relación de hiponimia, en el caso de los verbos llamamos a esta relación "troponimia". Esto es así porque se considera que las distinciones de modo son las más importantes para diferenciar un hipónimo verbal de su hiperónimo.

Existen otras dos relaciones en Wordnet de implicación entre verbos. La de oposición es una relación basada en un proceso morfológico en la mayoría de los casos. Es una relación compleja y, dentro de la oposición, se aplica a uno de sus miembros.

Otro tipo de relación entre dos conceptos es la causativa donde tenemos un concepto que será el causativo (ej: *give*) y otro que será el resultativo (ej: *have*). Siempre tendremos pares que estarán lexicalizados, de tal forma que la relación causativa es heredada por los sinónimos de los miembros del par haciendo que la relación no sea a nivel de palabras sino de conceptos. Siendo así, el concepto {*learn*, *acquire*, *knowledge*} tendrá como como sinónimos causativos a {*teach*, *instruct*, *educate*}.

Dependiendo del tipo de verbos que se quieran relacionar será más adecuado usar un tipo de relación u otra. Las relaciones de troponimia se usarían mejor para verbos que expresen movimiento, creación, consumo, comunicación o competición. Cuando tenemos verbos de cambio y de estado la relación más adecuada será la de oposición. La relación causativa se suele utilizar para verbos que implican movimiento.

3.5 RELACIONES SEMANTICAS

Conceptos de significado y significante

Dentro de la lingüística se podría definir la semántica como la disciplina que estudia el significado de las palabras. Podríamos decir que "significado" es la imagen mental que se forma en nuestra mente y que asociamos de manera estable a un sonido o imagen que será el "significante".

El significante de una palabra puede ser acústico o visual, algo material en cualquier caso. Se puede percibir por los sentidos al tener una naturaleza física. Es lo que oímos la hablar o lo que vemos al leer. Nos permite poder pensar de una manera fónica en una palabra sin pronunciarla: p-e-l-o-t-a.

El significado por su parte es la idea, el concepto formado en nuestra mente para una determinada palabra. De esta forma, a la sucesión de sonidos que forman la palabra p-e-l-o-t-a la mayoría de personas de habla hispana asocia una imagen parecida a esta:

Se podría decir que existen dos tipos de significado:

- **Significado denotativo**: sería el significado que encontraríamos en un diccionario al buscar una palabra. Es un significado formal y común para los hablantes de una lengua. Por ejemplo, según el diccionario de la RAE (Real Academia Española):

  Pelota: Bola de materia elástica que le permite botar y que se usa en diversos juegos y deportes.

- **Significado connotativo**: es un significado de tipo subjetivo, muy dependiente del contexto y que puede ser utilizado por menos hablantes o incluso por uno solo. Por ejemplo, la palabra “campo” para algunas personas puede significar (connotar) “naturaleza” y para otras “trabajo”, “calor”…

Generalmente las palabras tienen su parte denotativa y objetiva que viene complementada por su parte connotativa o subjetiva.

La poesía y en general el lenguaje literario, usa mucho la connotación para expresar ideas, emociones, estados de ánimo…Los significados denotativos no suelen variar sustancialmente con los cambios de época o cultura. Los significados connotativos sí se ven alterados por estos cambios de ciclo.

Tipos de relaciones semánticas

Entendemos como relaciones semánticas aquellas que se dan entre dos palabras atendiendo a su significado.

Dentro de estas, podemos destacar:

Sinonimia: esta relación se da cuando tenemos dos o más significantes para un mismo significado. Son palabras fonéticamente distintas pero que pertenecen gramaticalmente a la misma categoría. Son las palabras de significado similar. Ejemplos: pelota y balón, coche y vehículo, cielo y firmamento, etc…

Antonimia: es la relación que se da cuando dos palabras tienen significados opuestos o excluyentes. Ejemplos: vivo y muerto, enfermo y sano, bueno y malo, etc…

Homonimia: es la relación en la que tenemos varios significados para un mismo significante que proviene de orígenes distintos. Ejemplos: Llama como animal y llama de una vela, bello de belleza y vello como pelo del cuerpo, etc…

Se dividen en dos categorías:

Homógrafas: aquellas que se escriben igual.

Homófonas: aquellas con la misma pronunciación.

No se debe confundir el concepto de homonimia, que ya hemos visto que es una relación entre dos o más palabras, y el de *polisemia*, que es la posesión de dos o más significados por parte de una palabra de origen único, como por ejemplo: “sierra” como herramienta para cortar o “sierra” como conjunto de montañas.

Hiperonimia: es una relación que se da al tener un término que abarca un concepto más general que otro que es más específico. Ejemplos: vivienda y casa, ropa y pantalón, deporte y ciclismo, etc…

Hiponimia: relación en la que tenemos una palabra que concreta el significado de otra con un significado más general. Ejemplos: lunes y día, clavel y flor, verde y color, etc…

Holonimia: se da entre palabras en la que una representa el concepto del todo y otra el de una parte. Ejemplos: coche y volante, cuerpo y brazo, casa y salón, etc…

Meronimia: relación que se da cuando una palabra representa una parte dentro de otra de significado total. Ejemplos: diente y boca, sillín y bicicleta, país y continente, etc…

Además de los tipos de relaciones semánticas vistas, vamos a presentar otros conceptos que se usan en el desarrollo de este proyecto y que es interesante conocer:

Derivación: procedimiento mediante el cual se forman nuevas palabras mediante el uso de morfemas para modificar su significado. Los morfemas se dividen en prefijos, si van al principio del término, o sufijos si van al final. Aplicando este proceso podemos llegar a diferentes tipos de derivados:

- **Derivados populares**: surgen en la lengua hablada con el paso del tiempo a lo largo de siglos a partir de términos latinos o griegos que se han ido introduciendo directa o indirectamente de otros idiomas.
- **Derivados cultos**: los que aparecen dentro del lenguaje científico o de determinada rama de conocimiento. Se dividen a sus vez en:

- **Cultismos**: provienen de una de las denominadas lenguas clásicas y han sido introducidas en el idioma por exigencias culturales desde la literatura, la ciencia, la filosofía, la música, etc…No han sufrido las transformaciones habituales de los términos populares.

- **Tecnicismos**: aparecen normalmente de términos griegos propios de una industria, oficio, arte o ciencia y tienen un sentido concreto y determinado.

- **Neologismos**: son términos que aparecen para referirse a conceptos nuevos. Pueden ser inventados o crearse a partir de otros términos.

Dentro del contexto de Wordnet, existen dos tipos de relación que también estudiaremos en este proyecto:

- **Dominio** (de un *synset*): nos dice si un determinado *synset* pertenece a un dominio de Wordnet, entendiendo este como un campo de conocimiento concreto.

- **Miembro** (de un dominio): sería la relación opuesta a la anterior en la que nos indica si un determinado dominio tiene como miembro a elementos de un *synset* concreto.

En este trabajo se analizarán ocho tipos de estas relaciones entre palabras. Se buscarán cuando dos palabras son antónimas, hiperónimas, hipónimas, holónimas, merónimás, derivadas, cuál es su dominio en Wordnet y si son miembro de un dominio.

4.HERRAMIENTAS

4.1 Lenguaje de programación JAVA

Para implementar el software de análisis que se ha utilizado en el proyecto se ha escogido el JAVA (Sun Microsystems) como lenguaje de programación. La elección de este lenguaje se debe a varias razones:

- Es un lenguaje orientado a objetos que ofrece técnicas de desarrollo siguiendo este paradigma de programación y posibilita la reutilización de software.

- JAVA es un lenguaje simple que ofrece posibilidades potentes descartando algunas de las características más confusas de otros lenguajes como C++ . Algunas de las características que hacen más sencillo su uso son el no permitir la sobrecarga de operadores, el tener un modo automático para asignar y liberar memoria (recolector de basura), la existencia de la clase String que permite un mejor manejo de los arrays y eliminar la aritmética de punteros que se usaba en C y C++.

- JAVA es un lenguaje solido que no quiebra fácilmente ante errores de programación. Elimina la posibilidad de usar “atajos” que se permitían en C y C++ y que eran fuente constante de errores. No se permite, por ejemplo, hacer conversiones forzosas (cast) de enteros a punteros ni escribir en áreas arbitrarias de memoria.

- Es un lenguaje interpretado, donde cada programa en código fuente genera un archivo de tipo bytecode (código de bytes) que podrá ser interpretado en distintas máquinas en tiempo de ejecución según la máquina virtual que se esté utilizando.

- JAVA es un lenguaje portable. Implementa estándares de portabilidad que van más allá de la portabilidad básica que ofrecería cualquier arquitectura independiente. Por ejemplo, los enteros (int) tendrás siempre 32 bits en complemento a 2. Las operaciones aritméticas funcionaran igual independientemente de la plataforma que estemos utilizando ya que los tipos estándar como int o float se encuentran implementados de la misma manera en todas las máquinas.

4.2 Entorno de desarrollo ECLIPSE

Para la implementación del software necesario en el desarrollo de este proyecto se escogió como IDE (Integrated Development Environment) al entorno de desarrollo Eclipse.

La elección de Eclipse sobre otras opciones vino motivada por las siguientes características de este entorno:

- Es una plataforma de programación, desarrollo y compilación potente que abarca desarrollos de diferentes tipos como los hechos en C++, programas realizados en java o la creación de sitios web.

- Es un entorno de código abierto, integrado y multiplataforma. Fue creado por OTI (Object Technology International) para reemplazar a Visual Age. Al comienzo fue un proyecto de IBM Canadá. En Noviembre de 2001 se formó un consorcio para comenzar a desarrollar Eclipse con código abierto y en 2003 apareció la fundación que sería independiente de IBM. Inicialmente fue liberado con una licencia de tipo Common Public License y después se cambió a la Eclipse Public License, ambas de software libre, aunque según la Free Software Foundation son licencias incompatibles con la Publica General de GNU.

- El entorno de trabajo de Eclipse está formado por diferentes perspectivas. Estas son distintas ventanas de trabajo que se relacionan entre sí y que permiten cada una realizar una tarea diferente de una manera sencilla.

- Proporciona ayuda para la creación de proyectos facilitando la implementación de código, la documentación, el uso de ficheros de configuración o desplegando de manera adecuada el árbol de directorios.

- Contiene un depurador de código potente y de uso sencillo a la vez. Entrando en la perspectiva de depuración (debug) podemos ir siguiendo la ejecución del programa con gran detalle, haciendo uso de puntos de ruptura (breakpoints) a los que podemos aplicar condiciones de salto y seguir el valor de variables y expresiones del programa que podremos modificar a la vez que depuramos.

- Eclipse ofrece la posibilidad de instalar gran cantidad de *plugins* para la realización concreta de algunas tareas o para trabajar con ciertos lenguajes. Los hay de muy diferentes tipos, tanto gratuitos como de pago o con diferentes tipos de licencia. Pueden estar desarrollados por Eclipse o bien haber sido desarrollados por terceros.

4.3 Mysql

MySQL es un gestor de base de datos implementado en C/C++. Es una aplicación que nos permite manejar, crear y gestionar bases de datos. Cuenta con una licencia de tipo GPL (General Public License).

Algunas de las características de MySQL son:

- Posibilidad de trabajar con gran cantidad de datos en cada columna pudiendo ser estos de una amplia cantidad de tipos.
- Posibilidad de usar hasta 32 índices en tablas diferentes.
- Gestor robusto y de gran velocidad.
- Ofrece una amplia posibilidad de portabilidad entre sistemas, pudiéndose usar en distintas plataformas y sistemas operativos.
- Cada base de datos tiene tres archivos: uno de estructura, una para datos y otro par índices.
- Su implementación multihilo posibilita realizar tareas multiprocesador.
- Importante nivel de seguridad con un buen sistema de gestión de usuarios y contraseñas.
- Necesidades bajas de requerimientos de las máquinas en la que vaya a ser ejecutado ya que tiene un bajo consumo de los mismos.
- Fácilmente instalable y configurable.

4.4 MySQL Workbench

MySQL Workbench es una herramienta que permite trabajar de forma más sencilla con bases de datos de tipo MySQL tanto para su diseño como para la elaboración de consultas u otras funciones que puedan ser necesarias al enfrentarnos a un almacenamiento de datos de este tipo.

Algunas de las características interesantes de este entorno son:

- Es un entorno gratuito en su versión Community libremente distribuida con licencia GPL.
- Es multiplataforma. Se puede usar en Windows, GNU/Linux y Mac.
- Permite la representación de las tablas de forma visual , claves foráneas, funciones almacenadas y procedimientos que hace que la información sea rápidamente asimilable.

- Permite importar archivos SQL y crear diagramas E-R.
- Tiene una herramienta para hacer migraciones de bases de datos de otros tipos a MySQL.
- Es un entorno de instalación fácil y de configuración básica sencilla.
-
- Existe bastante documentación para su uso.

5. DISEÑO

5.1 Datos de entrada

Para iniciar el trabajo, partimos de una colección de más de 800 archivos en formato PDF que trataban diversos temas relacionados con la sordera genética y la perdida y deficiencia auditiva. Estos documentos se encontraban almacenados en CD y distribuidos en diversas carpetas y subcarpetas. Queríamos pasar la información a archivos de texto plano. Para ello se desarrolló un programa en java que iba recorriendo todo el sistema de archivos e iba extrayendo el texto de cada pdf para almacenarlo como archivo de texto plano. El programa recorría el file system (que se había copiado del CD al disco duro del equipo utilizado para la realización del proyecto) e iba generando archivos de tipo ".txt" cuyo nombre era un número del uno en adelante (hasta la totalidad de ficheros) para luego poder tratarlos mejor en los pasos posteriores. Para la extracción del texto se hizo uso de la librería Apache PDFBox dado que era de uso sencillo y estaba bien valorada en cuanto a su desempeño. Hubo que descartar algún pdf ya que daba problemas de encriptado. Se intentó acceder a su contenido haciendo uso de alguna de las opciones que ofrece PDFBox pero el resultado no fue satisfactorio. En cualquier caso no supuso una gran pérdida de información y la colección final fue considerada más que suficiente para trabajar ya que quedó conformada por 756 archivos con extensión ".txt" con diferentes tamaños y numerados del 1 al 756.

Un tema aparte a considerar fue la aparición de caracteres ilegibles en el resultado final de los archivos en formato ".txt". La gran mayoría de caracteres especiales contenidos en los archivos pdf se consiguió representar correctamente al pasar a texto plano. No obstante, seguían apareciendo algunos símbolos de difícil interpretación que no fue posible eliminar. Mirando un poco en los metadatos de los archivos pdf se veía que no estaban uniformemente codificados y que muchas veces, dentro de un mismo documento, se usaban codificaciones distintas. En algunos documentos ni siquiera fue posible encontrar la información de la codificación que usaban.

5.2 Migración de base de datos

Al proponerse el presente trabajo se me facilitaron dos bases de datos de tipo Access. La primera, que llamaremos 'bd_wordnet', contenía información disponible en dicha plataforma. Entre otras, podíamos obtener información sobre sustantivos y sus sinónimos (siempre en lengua inglesa), así como los identificadores asignados para cada uno de ellos. También podíamos acceder a diferentes tablas que nos daban, de haberla, la relación entre cada par de sustantivos. Siendo así, podíamos averiguar cuando dos sustantivos eran hiperónimos, hipónimos, holónimos, etc…y cuál era el padre y cuál el hijo dentro de la tupla. Cada grupo de sustantivos sinónimos se almacenan en Wordnet bajo un identificador llamado 'synset'. Este

identificador era el que aparecía en las tablas de relaciones y a partir del cual se podía acceder al término en cuestión tanto hijo como padre.

Una segunda tabla, a la que llamaremos 'bd_resultados', englobaba información sobre los resultados de un trabajo anterior que se realizó en el departamento de informática de la universidad Carlos III y en el que se pretendía evaluar la fiabilidad de ciertas cadenas de texto para establecer relaciones entre sustantivos. En dicho trabajo se evaluaban en concreto dichas cadenas para establecer si dos términos eran hiperónimos, hipónimos, holónimos, merónimos, antónimos, derivados, miembros o de pertenencia a un dominio.

De dicho trabajo rescatamos las cadenas intermedias obtenidas para cada tipo de relación estudiado y poder ver en qué número se encontraban en nuestra colección de textos sobre la enfermedad auditiva y para analizar posteriormente su fiabilidad como elementos que estableciesen uno de los ocho tipos de relaciones consideradas.

Como ya dijimos, ambas bases de datos se encontraba en formato Access. Se planteó la posibilidad de migrarlas a otro tipo de base de datos que pudiese hacer más ágil el manejo de dicha información y que dispusiese de algún entorno más amigable para trabajar con ella. Se tomó la decisión de llevar los datos a una base de datos de tipo MySQL dado que por la experiencia personal era de fácil manejo y relativamente rápida.

El proceso de migración planteó algunos problemas. La base de datos de Wordnet tenía un tamaño de unos 26MB. El proceso de migración se realizó a través de una herramienta que incorpora MySQL Workbench llamada Database Migration.

Previamente, para poder realizar la migración, tuvimos que configurar un ODBC para podernos conectar a la base de datos en Access. ODBC (Open DataBase Connnectivity) es el estándar que usa Windows (Microsoft) para conectarse a diferentes tipos de bases de datos. Para ello, la base de datos a la que deseamos conectarnos tendrá que disponer de un driver ODBC que permita la conexión. Se ha de crear un DNS (Data Source Name) como nombre de la conexión ODBC. Este será el nombre al que se hará referencia cuando se quiera hacer uso de dicha conexión desde otras aplicaciones. MySQL, Access o SQL Server son ejemplos de bases de datos que tienen su propio driver ODBC y que además viene ya instalado por defecto en los sistemas Windows.

A través de los diferentes pasos que propone la herramienta, se puede ir completando el proceso de una forma relativamente sencilla.

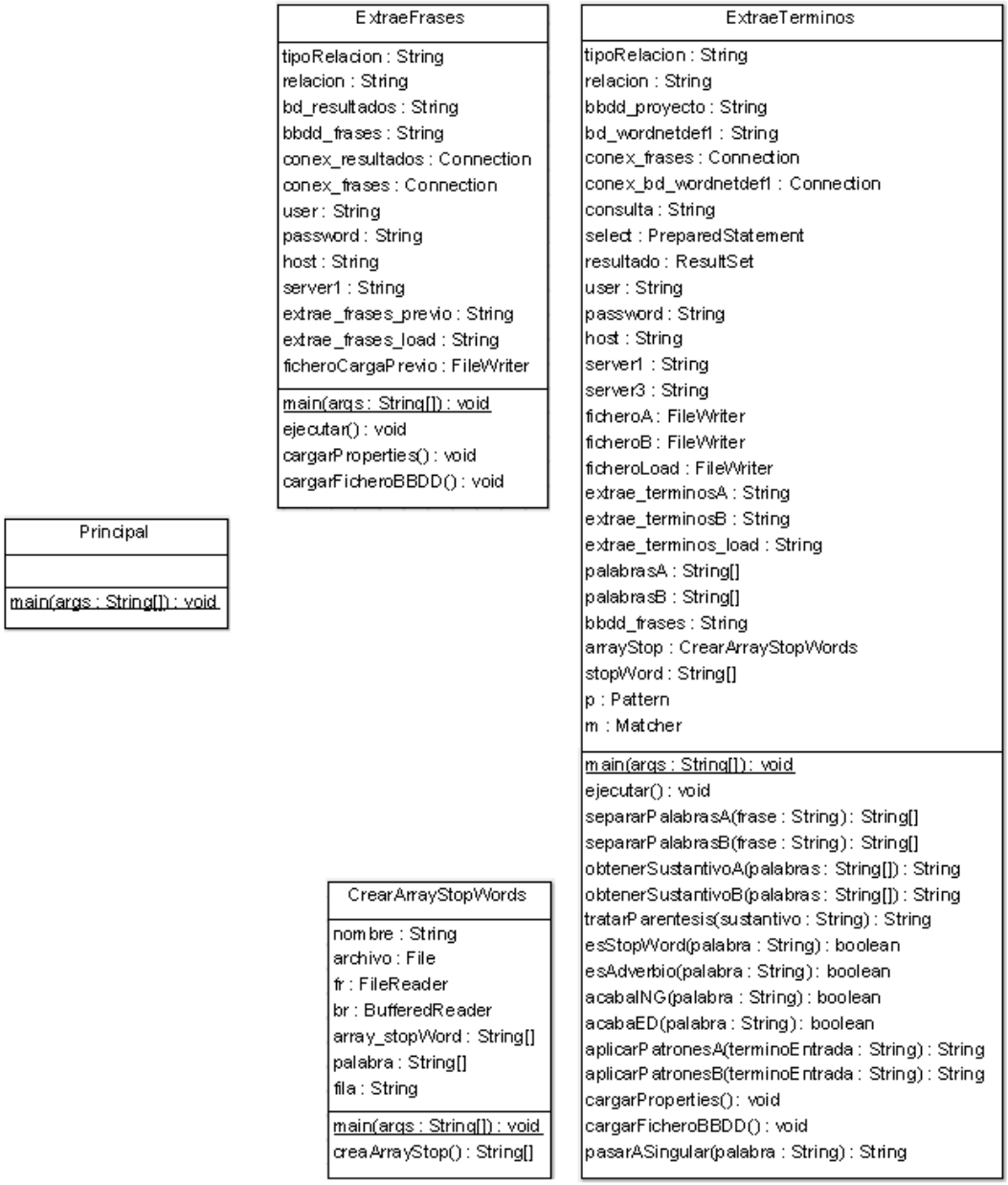
La base de datos de Wordnet no planteó problemas para ser migrada a MySQL, pero no así la base de datos 'bd_resultados', ya que está poseía más de 2GB de información y hacía que la herramienta de migración no pudiese manejarla.

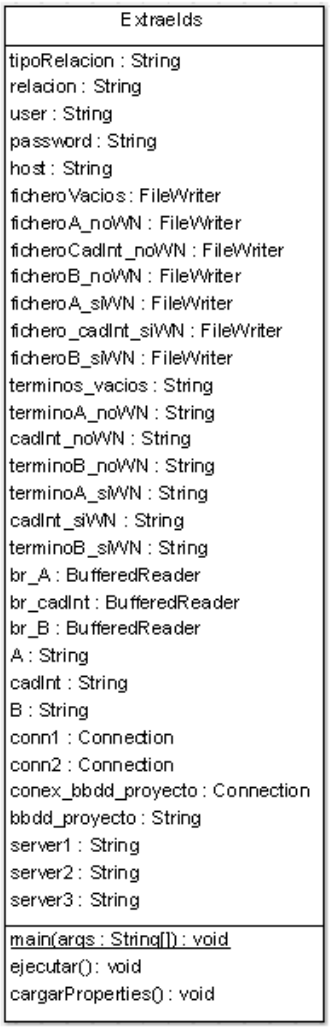
Ante esta situación se desarrolló un programa en Java que extraía los datos de la base de datos 'bd_resultados' en Access, los mandaba a un fichero y luego los cargaba en la base de datos destino en MySQL. El resultado fue satisfactorio.

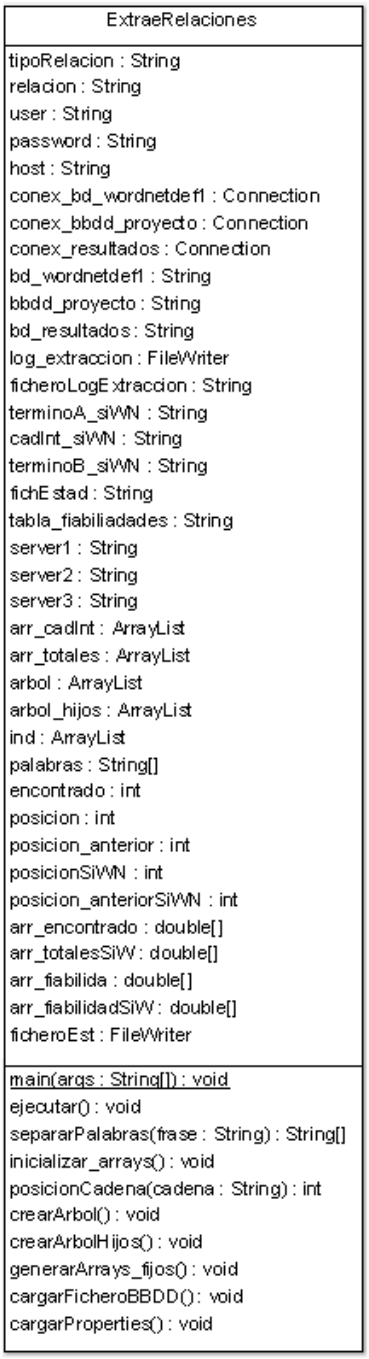
5.3 Diseño de clases

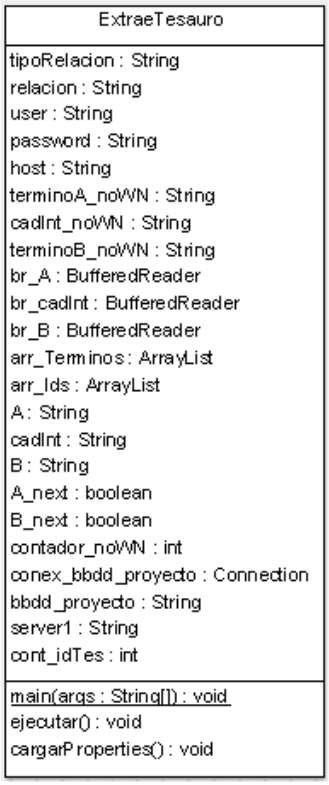
Una vez teníamos la colección de artículos sobre la sordera genética en texto plano y las dos bases de datos migradas a MySQL se planteó el diseño del software que realizaría las diferentes tareas que nos llevarían a los resultados buscados.

A continuación se hace un resumen de las clases creadas en este proyecto indicando brevemente su función y una descripción de sus principales métodos:




ExtraeIds
tipoRelacion : String
relacion : String
user : String
password : String
host : String
ficheroVacios : FileWriter
ficheroA_noWN : FileWriter
ficheroCadInt_noWN : FileWriter
ficheroB_noWN : FileWriter
ficheroA_siWN : FileWriter
fichero_cadInt_siWN : FileWriter
ficheroB_siWN : FileWriter
terminos_vacios : String
terminoA_noWN : String
cadInt_noWN : String
terminoB_noWN : String
terminoA_siWN : String
cadInt_siWN : String
terminoB_siWN : String
br_A : BufferedReader
br_cadInt : BufferedReader
br_B : BufferedReader
A : String
cadInt : String
B : String
conn1 : Connection
conn2 : Connection
conex_bbdd_proyecto : Connection
bbdd_proyecto : String
server1 : String
server2 : String
server3 : String
main(args : String[]) : void
ejecutar() : void
cargarProperties() : void


ExtraeRelaciones
tipoRelacion : String
relacion : String
user : String
password : String
host : String
conex_bd_wordnetdef1 : Connection
conex_bbdd_proyecto : Connection
conex_resultados : Connection
bd_wordnetdef1 : String
bbdd_proyecto : String
bd_resultados : String
log_extraccion : FileWriter
ficheroLogExtraccion : String
terminoA_siWN : String
cadInt_siWN : String
terminoB_siWN : String
fichEstad : String
tabla_fiabiliadades : String
server1 : String
server2 : String
server3 : String
arr_cadInt : ArrayList
arr_totales : ArrayList
arbol : ArrayList
arbol_hijos : ArrayList
ind : ArrayList
palabras : String[]
encontrado : int
posicion : int
posicion_anterior : int
posicionSiWN : int
posicion_anteriorSiWN : int
arr_encontrado : double[]
arr_totalesSiW : double[]
arr_fiabilida : double[]
arr_fiabilidadSiW : double[]
ficheroEst : FileWriter
main(args : String[]) : void
ejecutar() : void
separarPalabras(frase : String) : String[]
inicializar_arrays() : void
posicionCadena(cadena : String) : int
crearArbol() : void
crearArbolHijos() : void
generarArrays_fijos() : void
cargarFicheroBBDD() : void
cargarProperties() : void


ExtraeTesauro
tipoRelacion : String
relacion : String
user : String
password : String
host : String
terminoA_noWN : String
cadInt_noWN : String
terminoB_noWN : String
br_A : BufferedReader
br_cadInt : BufferedReader
br_B : BufferedReader
arr_Terminos : ArrayList
arr_Ids : ArrayList
A : String
cadInt : String
B : String
A_next : boolean
B_next : boolean
contador_noWN : int
conex_bbdd_proyecto : Connection
bbdd_proyecto : String
server1 : String
cont_idTes : int
main(args : String[]) : void
ejecutar() : void
cargarProperties() : void

CLASE: Principal.java

Descripción: Clase java que va llamando al resto para ejecutar la operativa correcta y para cada uno de los ocho tipos de relación que se tratan en el presente proyecto.

Métodos:

- **public static void main(String[] args) throws ClassNotFoundException,SQLException, IOException:** clase principal que va llamando al resto de clases para que ejecuten su funcionalidad.

CLASE: ExtraeFrases.java

Descripción: Clase java que se encarga de extraer las cadenas intermedias que tomamos de la columna 'cadena_intermedia' de la base de datos 'bd_resultados' de cada una de las ocho tablas de cada tipo de relación estudiada en este proyecto. Para cada tipo de relación (tabla) busca todas las cadenas intermedias en cada uno de los ficheros ".txt" que obtuvimos a partir de los ficheros PDF que se nos proporcionaron relacionados con el tema de la sordera genética. En este paso se extrae tanto la cadena intermedia como el texto que aparece antes y después de esta hasta un máximo de 30 caracteres.

Métodos:

- **public void cargarProperties(String tipoRelacion):** método encargado de cargar el fichero de configuración correspondiente al tipo de relación entre sustantivos que estamos tratando en ese momento.
- **public void ejecutar() throws IOException, ClassNotFoundException,SQLException:** método que va recorriendo los diferentes archivos de texto plano de los que tenemos que extraer las cadenas intermedias que se analizarán para cada tipo de relación entre sustantivos. Además de las cadenas intermedias coge los 30 caracteres a izquierda y derecha de esta para formar una frase que se analizará después en busca de posibles sustantivos a ambos lados de las cadenas. El resultado de esta búsqueda se almacena en fichero.

- **public void cargarFicheroBBDD() throws ClassNotFoundException,SQLException, IOException:** almacena en base de datos el fichero generado en el método ejecutar().

CLASE: ExtraeTerminos.java

Descripción: Clase java encargada de buscar el primer sustantivo que aparece tanto antes como después de la cadena intermedia. En esta clase se tomaron diversas decisiones para intentar obtener, en la medida de lo posible, solo sustantivos, dado que es lo que más nos interesaba para el estudio que intentamos hacer. Se intentó descartar adverbios, palabras vacías (stopword en inglés) como por ejemplo "about", "also" o "else", comparando con una lista de más de 500 de estas palabras. Se intentó pasar a singular la mayoría de plurales encontrados en las frases. Por último, se aplicaron también una serie de patrones para eliminar los casos en que la cadena intermedia candidata no lo era en realidad (por ejemplo aquellas frases en que la cadena intermedia era comienzo de frase) o para permitir solamente ciertos caracteres especiales.

Métodos:

- **public void cargarProperties(String tipoRelacion) :** método encargado de cargar el fichero de configuración correspondiente al tipo de relación entre sustantivos que estamos tratando en ese momento.

- **public String[] separarPalabrasA(String frase) :** método encargado de separar en palabras la frase (String) que tenemos a la izquierda de la cadena intermedia que estamos analizando en ese momento. También intenta unir las palabras que en los archivos de texto estuviesen cortadas por un salto de línea.

- **public String[] separarPalabrasB(String frase) :** método encargado de separar en palabras la frase (String) que tenemos a la derecha de la cadena intermedia que estamos analizando en ese momento. También intenta unir las palabras que en los archivos de texto estuviesen cortadas por un salto de línea.

- **public String obtenerSustantivoA(String[] palabras) :** método que obtiene el término a la izquierda de la cadena intermedia una vez se han descartado palabras de tipo "stop word", posibles adverbios, palabras acabadas en "ed" y palabras acabadas en "ing". Aplica también una serie de patrones para limpiar los términos de algunos caracteres especiales. Por último, hace un tratamiento que pasa a singular posibles plurales encontrados.

- **public String obtenerSustantivoB(String[] palabras) :** método que obtiene el término a la derecha de la cadena intermedia una vez se han descartado palabras de tipo "stop word", posibles adverbios, palabras acabadas en "ed" y palabras acabadas en "ing". Aplica también una serie de patrones para limpiar los términos de algunos caracteres especiales. Por último, hace un tratamiento que pasa a singular posibles plurales encontrados.

- **public boolean esStopWord(String palabra) :** método que comprueba si una palabra pertenece a un listado de más de 500 palabras que se denominan "stop word".

- **public boolean esAdverbio(String palabra) :** método que comprueba si una palabra es mayor de seis letras y acaba en “ly”.

- **public boolean acabaED(String palabra):** método que comprueba si una palabra finaliza en “ed”.

- **public boolean acabaING(String palabra) :** método que comprueba si una palabra finaliza en “ing”.

- **public String pasarASingular(String palabra) :** método que intenta pasar posibles palabras plurales a singulares.

- **public String aplicarPatronesA(String terminoEntrada) :** método que aplica ciertos patrones mediante expresiones regulares a las palabras tratadas a la izquierda de la cadena intermedia para limpiarlas de ciertos caracteres especiales.

- **public String aplicarPatronesB(String terminoEntrada) :** método que aplica ciertos patrones mediante expresiones regulares a las palabras tratadas a la derecha de la cadena intermedia para limpiarlas de ciertos caracteres especiales.

- **public void ejecutar() throws IOException, ClassNotFoundException,SQLException:** método que recupera de base de datos las frases que vamos a analizar y va llamando a los métodos anteriores para terminar creando tres ficheros, dos con los términos a izquierda y derecha respectivamente de la cadena intermedia que se analiza en cada caso y otro fichero que se cargará en base de datos con dichos términos.

- **public void cargarFicheroBBDD() throws ClassNotFoundException, SQLException:** método que carga en base de datos los términos obtenidos en el método ejecutar().

CLASE: CrearArrayStopWords.java

Descripción: Clase java que crea un array de palabras de tipo ‘stop word’ a partir de la lectura de un fichero en el que tendremos nuestra lista de palabras de este tipo en lengua inglesa.

Métodos:

- **public String[] creaArrayStop():** método que va leyendo de fichero y generando el array de palabras de tipo ‘stop word’.

CLASE: ExtraeIds.java

Descripción: En esta clase buscamos en la base de datos 'bd_wordnet' con información de Wordnet los identificadores de los sustantivos que allí se encuentren. Estos identificadores serán usados después para establecer qué relaciones encontramos entre cada par de sustantivos relacionados por cada cadena intermedia. Los sustantivos o términos que no son encontrados en la base de datos los guardamos para poder asignar posteriormente una probabilidad de que la relación entre dicho término y su par sea una de las ocho estudiadas en este proyecto. Estos términos son los que, en principio, serían la base para formar un tesauro dentro del dominio de conocimiento de la sordera genética. En este proyecto se trata solo con términos y no con posibles sintagmas nominales. Esto hace que la búsqueda que hace esta clase sea solo de términos individuales.

Métodos:

- **public void cargarProperties(String tipoRelacion):** método encargado de cargar el fichero de configuración correspondiente al tipo de relación entre sustantivos que estamos tratando en ese momento.

- **public void ejecutar() throws IOException, SQLException,ClassNotFoundException:** método que extrae de la base de datos de Wordnet el número que identifica a un sustantivo, pudiendo aparecer varias veces si tiene diferentes significados. En caso de no obtener identificador almacenamos dichos términos para tratarlos posteriormente.

CLASE: ExtraeRelaciones.java

Descripción: Esta clase busca, para cada par de sustantivos de los cuales tenemos su identificador Wordnet, el tipo o tipos de relación que genera la cadena intermedia que los relaciona dentro de los ocho tipos de relación estudiadas. Para encontrar esta posible relación buscamos en la base de datos de Wordnet los pares de sustantivos (identificadores) que aparecen en cada una de las tablas que muestran relaciones entre pares de sustantivos. En este paso obtendremos, para cada tipo de relación, un listado de las cadenas intermedias estudiadas y para cada una aparecerá las veces que aparecía en nuestra colección de textos, las veces que estableció un tipo de relación valida y el porcentaje de fiabilidad por lo tanto para establecer ese tipo de relación.

Métodos:

- **public void cargarFicheroBBDD() throws ClassNotFoundException, SQLException:** método que carga en base de datos el fichero que se ha ido generando en la búsqueda de relaciones entre términos. Este fichero contiene las distintas cadenas intermedias que establecieron relación entere términos que se encontraban en Wordnet, el número de veces que aparecían en los textos, el número de veces que establecían una relación valida del tipo estudiado en cada

caso y una asignación de probabilidad de que esa cadena establezca el tipo de relación que se está tratando.

- **public void cargarProperties(String tipoRelacion):** método encargado de cargar el fichero de configuración correspondiente al tipo de relación entre sustantivos que estamos tratando en ese momento.

- **public void crearArbol(String relacion) throws SQLException,ClassNotFoundException:** método que, tras consultar en la base de datos de Wordnet, crea un ArrayList<ArrayList> para cada tipo de relación estudiada en la que cada posición guarda una entrada con el hijo en primera posición seguido de todos sus padres.

- **public void crearArbolHijos():** método que crea un ArrayList<String> en los que almacena solo los hijos que aparecen en la estructura que crea el método *crearArbol* anterior.

- **public void inicializar_arrays():** método que inicializa los arrays de 'encontrados' y 'fiabilidades' en los que se almacenarán las veces que una cadena intermedia establece relación valida y la fiabilidad calculada posteriormente para la misma para establecer un tipo de relación determinada.

- **public int posicionCadena(String cadena) throws NumberFormatException,IOException:** método que devuelve la posición que ocupa una cadena intermedia dentro de la lista de cadenas intermedias estudiadas para cada tipo de relación y que se almacenaron en un array para manejarlas más fácilmente.

- **public String[] separarPalabras(String frase):** método que divide en palabras separadas por espacios cada línea de los ficheros que almacenan los sustantivos encontrados en Wordnet y sus identificadores. La primera palabra serán el sustantivo estudiado y las siguientes sus identificadores(números enteros).

- **public void ejecutar() throws IOException:** método que coge los identificadores de los términos encontrados en Wordnet (tanto a izquierda como a derecha de la cadena intermedia considerada) y busca en la base de datos de Wordnet si establecen relación del tipo de la estudiada en cada momento.

CLASE: ExtraeTesauro.java

Descripción: Esta clase toma los pares de palabras para los cuales al menos una de ellas no tenía identificador en Wordnet y les asigna una probabilidad de que entre ellas se establezca uno de los ocho tipos de relación estudiados en este proyecto.

Métodos:

- **public void cargarProperties(String tipoRelacion):** método encargado de cargar el fichero de configuración correspondiente al tipo de relación entre sustantivos que estamos tratando en ese momento.
- **public void ejecutar() throws IOException, SQLException,ClassNotFoundException:** método que asigna un identificador a cada palabra no encontrada en Wordnet y que establece una probabilidad que de que cada par de sustantivos a ambos lados de cierta cadena intermedia tengan uno de los ocho tipos de relación estudiados.

6. RESULTADOS

En este capítulo se presentan los resultados del proceso de análisis de los textos tratados. Por cada uno de los ocho tipos de relación estudiados entre sustantivos se muestran los resultados obtenidos en los subcapítulos (6.1-6.8). En el subcapítulo 6.9, denominado “miscelánea”, se presentan algunos resultados que nos parecen de interés tras elaborar este proyecto y que intentan salirse de la mera exposición numérica.

Por cada tipo de relación se podrán ver los siguientes datos en los subcapítulos (6.1-6.8):

- **Cadenas encontradas**: número total de cadenas intermedias encontradas en los textos analizados buscando, para cada tipo de relación, 999 cadenas. Algunas de estas cadenas son comunes a dos o más tipos de relación.

- **Cadenas que relacionan un par de términos ambos en Wordnet:** número de cadenas intermedias encontradas que aparecen en los textos para cada tipo de relación y cuyos términos extraídos a izquierda y derecha de la misma se han encontrado en la base de datos proporcionada de Wordnet.

- **Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet:** número de cadenas intermedias que se han encontrado en los textos para cada tipo de relación y en las que tras el proceso de extracción de términos a izquierda y derecha de la misma, al menos uno de ellos no se encontraba en la base de datos de Wordnet.

- **Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío:** número de cadenas intermedias que se han encontrado en los textos para cada tipo de relación en y en las que no se ha podido extraer alguno de los términos a izquierda o derecha de la misma atendiendo a diferentes criterios de exclusión.

- **Relaciones encontradas:** número de relaciones encontradas para cada tipo de relación cuando teníamos una cadena intermedia cuyos términos a izquierda y derecha se encontraron en la base de datos de Wordnet.

Tras estos datos encontraremos, para cada tipo de relación, tres tablas que representan:

- **Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:** muestra las 25 cadenas intermedias que han aparecido más veces en los textos para el tipo de relación concreto que se está estudiando. Se presentan las apariciones totales, independientemente de los términos que se hayan podido extraer a su izquierda o a su derecha. Se han sacado de la base de datos que se ha elaborado en este proyecto. La tabla tendrá el siguiente aspecto:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240

La columna "*cadena_ intermedia*" muestra la cadena intermedia considerada y la columna "*apariciones_totales*" el número total de veces que aparece en los textos tratados.

- **25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:** muestra las 25 primeras cadenas intermedias que se encontraron y en las que sus términos a izquierda y derecha aparecían en la base de datos de Wordnet. De estos se muestra la cifra de los que establecían el tipo de relación estudiado según Wordnet y por lo tanto una fiabilidad estimada de establecer ese tipo de relación para esa cadena intermedia concreta. Los resultados se muestran en orden decreciente de dicha fiabilidad.

  La tabla tendrá un aspecto similar a este:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
v	3	1	0.333333
and excessive	3	1	0.333333
as opposed to	17	2	0.117647

La columna "*cadena_intermedia*" muestra la cadena considerada. La columna "*num_apariciones*" es el total de veces que aparece esa cadena y tiene términos de Wordnet a sus lados izquierdo y derecho. La columna "*num_relaciones*" presenta el total de relaciones encontradas en Wordnet para el tipo de relación estudiado para esa cadena y el par de términos que relaciona. La columna "*fiabilidad*" es una estimación de la probabilidad de la cadena de establecer la relación tratada. Esta última se obtiene dividiendo el valor de la columna "*num_relaciones*" entre la columna "*num_apariciones*".

- **25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:** muestra los 25 primeros pares de términos encontrados y relacionados por una determinada cadena intermedia en los que al menos uno de ellos no se encontró en la base de datos de Wordnet. Se han ordenado en orden decreciente de probabilidad de que tuviesen el tipo de relación estudiado en base a la cadena intermedia que los asocia. La tabla tendrá un aspecto similar al siguiente:

terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
0.67	v	0.61	17633	326	0.333333
genescan	v	3.7	12750	1114	0.333333
genotyper	v	3.7	21137	1114	0.333333

La columna "*terminoA*" es el término a la izquierda de la cadena intermedia. La columna "*cadena_intermedia*" es la cadena estudiada en cada caso. La columna "*terminoB*" es el término a la derecha de la cadena. La columna "*id_hijo*" es un identificador que se ha asignado a ese término a la izquierda de la cadena y que se guarda en una tabla de la base de datos creada en el proyecto. La columna "*id_padre*" es el identificador asignado al término de la derecha. La columna "*fiabilidad*" es la probabilidad de que los términos de esa fila tengan la relación estudiada.

6.1 Antónimos

Cadenas encontradas: 1177862
Cadenas que relacionan un par de términos ambos en Wordnet: 418033
Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet: 455552
Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío: 304277
Relaciones encontradas: 973

Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240
to	63535
a	61275
of the	41447
with	41385
for	38183
is	34758
by	31883
et	28726
in the	26372
was	24240
are	21002
from	20888
as	18894
or	15649
on	15466
an	13380
not	12186
to the	10059
2	8365
and the	7562
also	6771
with the	6143

Tabla 1. Antónimos: Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición.

25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
v	3	1	0.333333
and excessive	3	1	0.333333
as opposed to	17	2	0.117647
or	4902	240	0.0489596
or the	213	6	0.028169
7	663	12	0.0180995
nor	63	1	0.015873
when the	136	2	0.0147059
and not	77	1	0.012987
and	38072	416	0.0109267
than	1196	13	0.0108696
rather than	184	2	0.0108696
as well as	429	4	0.00932401
3	780	7	0.00897436
and one	117	1	0.00854701
long	267	2	0.00749064
and a	764	4	0.0052356
and the	3170	12	0.00378549
but not	283	1	0.00353357
to	19377	61	0.00314806
with an	400	1	0.0025
2	1246	3	0.0024077
an	3888	9	0.00231481
as	6440	14	0.00217391
then	559	1	0.00178891

Tabla 2. Antónimos: 25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados.

25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:

terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
0.67	v	0.61	17633	326	0.333333
genescan	v	3.7	12750	1114	0.333333
genotyper	v	3.7	21137	1114	0.333333
navigator	v	1.5.3	23307	50463	0.333333
patient	v	nf1	95	47	0.333333
premature	and excessive	vessel	1868	10871	0.333333
profile	and excessive	extracellular	203	2136	0.333333
spss	v	12.0	36943	34942	0.333333
catalytic	as opposed to	structural	5510	927	0.117647
cell	as opposed to	osteoprogenitor	360	19094	0.117647
complex	as opposed to	mice	1847	81	0.117647
concept	as opposed to	mechan	27547	17713	0.117647
conformation	as opposed to	curvedof	3343	31012	0.117647
constitutionalnf2mutation	as opposed to	somatic	30138	1176	0.117647
dosage	as opposed to	physical	3975	386	0.117647
endogene	as opposed to	transgene	30076	3619	0.117647
environment	as opposed to	dim	2313	8756	0.117647
gjb2	as opposed to	loss	1339	486	0.117647
mutation	as opposed to	somatic	79	1176	0.117647
premature	as opposed to	constitutive	1868	1593	0.117647
schwannoma	as opposed to	neurofibroma	4129	720	0.117647
sequence	as opposed to	highly	193	1193	0.117647
sequence	as opposed to	nf-1	193	6903	0.117647
site	as opposed to	nf-1	34	6903	0.117647
smnd7	as opposed to	function	3611	989	0.117647

Tabla 3. Antónimos: 25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado.

6.2 Derivados

Cadenas encontradas: 1211481
Cadenas que relacionan un par de términos ambos en Wordnet: 448468
Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet: 479218
Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío: 283795
Relaciones encontradas: 58

Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240
in	110000
to	63535
a	61275
of the	41447
with	41385
for	38183
is	34758
that	32887
by	31883
from	20888
as	18894
at	18738
or	15649
on	15466
an	13380
to the	10059
which	9231
and the	7562
for the	6250
type	5725
of a	5675
but	5526

Tabla 4. Derivados: Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición.

25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
whose	192	1	0.00520833
and a	764	1	0.0013089
which	2714	2	0.00073692
and the	3170	2	0.000630915
different	1831	1	0.00054615
as	6440	3	0.000465839
from the	2255	1	0.000443459
or	4902	2	0.000407997
from	8274	3	0.000362582
and	38072	13	0.000341458
that	9819	2	0.000203687
to	19377	3	0.000154823
with	13904	2	0.000143843
the	76680	10	0.000130412
in	46558	6	0.000128872
a	22963	2	0.0000870966
of the	19170	1	0.0000521648
of	78848	3	0.0000380479
addiction	1	0	0
activities that awaits the	0	0	0
4wd this 1989 toyota	0	0	0
4	384	0	0
3 male	0	0	0
your	50	0	0
2007 so owner	0	0	0

Tabla 5. Derivados: 25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados.

25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:

terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
13q12	whose	order	15544	11233	0.00520833
155-1,5	whose	deletion	56048	202	0.00520833
35delg	whose	biological	10520	2302	0.00520833
a40v	whose	position	16381	858	0.00520833
anewborn	whose	weight	56051	8987	0.00520833
apologize	whose	essential	11083	2901	0.00520833
aso	whose	target	1607	3067	0.00520833
asp54rbx	whose	carboxylate	41767	20106	0.00520833
bas-rhin	whose	birth	56050	1474	0.00520833
berber	whose	nry	41226	13007	0.00520833
boy	whose	tibial	4354	241	0.00520833
cas	whose	diagnosis	200	3669	0.00520833
children	whose	clinical	808	248	0.00520833
children	whose	mother	808	2737	0.00520833
component	whose	microt	1874	17581	0.00520833
condition	whose	molecular	1917	1461	0.00520833
cx30	whose	expression	8949	427	0.00520833
deaminase	whose	expression	44384	427	0.00520833
deletion	whose	proximal	202	1632	0.00520833
diseas	whose	common	1802	2075	0.00520833
disorder	whose	clinical	2033	248	0.00520833
domain	whose	structural	169	927	0.00520833
dsb	whose	dna-pkcs	4748	14972	0.00520833
e3209	whose	susceptible	29318	7014	0.00520833
elegan	whose	genome	5314	2140	0.00520833

Tabla 6. Derivados: 25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado.

6.3 Hiperónimos

Cadenas encontradas: 1491484
Cadenas que relacionan un par de términos ambos en Wordnet: 544179
Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet: 582173
Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío: 365132
Relaciones encontradas: 5917

Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240
in	110000
to	63535
a	61275
of the	41447
with	41385
for	38183
is	34758
that	32887
by	31883
in the	26372
was	24240
are	21002
from	20888
as	18894
at	18738
or	15649
on	15466
an	13380
cells	12513
not	12186
1	12128
this	11911

Tabla 7. Hiperónimos: Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición.

25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
strategic	3	2	0.666667
is a complex	8	4	0.5
the title	3	1	0.333333
title	4	1	0.25
is defined as	8	2	0.25
land	4	1	0.25
food	9	2	0.222222
your	50	9	0.18
of your	6	1	0.166667
and spontaneous	7	1	0.142857
grey	33	4	0.121212
and other	229	26	0.113537
of or	9	1	0.111111
time the	9	1	0.111111
or other	60	6	0.1
for that	10	1	0.1
fusion	127	12	0.0944882
and his	37	3	0.0810811
gender	13	1	0.0769231
soft	45	3	0.0666667
physical	122	8	0.0655738
chemical	47	3	0.0638298
medical	96	6	0.0625
you	35	2	0.0571429
an active	36	2	0.0555556

Tabla 8. Hipéronimos: 25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados.

25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:

terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
ctf	strategic	forum	4117	34644	0.666667
nmdar	strategic	position	9946	858	0.666667
arhi	is a complex	disorder	5390	2033	0.5
deafness	is a complex	genetic	1977	985	0.5
disease	is a complex	autoimmune	1708	12360	0.5
ear	is a complex	bony	5355	1513	0.5
egfr	is a complex	system	1761	3624	0.5
iugr	is a complex	disease	10507	1708	0.5
mice	is a complex	trait	81	8083	0.5
neurofibromin	is a complex	protein	546	274	0.5
op18	is a complex	issue	2992	3648	0.5
ps-eva	is a complex	disease	5643	1708	0.5
tumorigenesi	is a complex	multistep	6616	4081	0.5
change	the title	antonio	2045	16549	0.333333
justify	the title	essay	43309	38009	0.333333
mutation	the title	allude	79	11816	0.333333
profilin	the title	universal	843	2471	0.333333
change	title	antonio	2045	16549	0.25
justify	title	essay	43309	38009	0.25
loss	is defined as	bilateral	486	2001	0.25
loss	is defined as	profound	486	2036	0.25
mutation	title	allude	79	11816	0.25
print	title	additional	7494	460	0.25
profilin	title	universal	843	2471	0.25
resistance	land	molecular	1387	1461	0.25

Tabla 9. Hiperónimos: 25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado.

6.4 Hipónimos

Cadenas encontradas: 1568523
Cadenas que relacionan un par de términos ambos en Wordnet: 569305
Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet: 611754
Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío: 387464
Relaciones encontradas: 8668

Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240
in	110000
to	63535
a	61275
of the	41447
with	41385
for	38183
is	34758
that	32887
by	31883
in the	26372
were	24872
was	24240
are	21002
from	20888
as	18894
at	18738
or	15649
on	15466
an	13380
not	12186
1	12128
this	11911

Tabla 10. Hipónimos: Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición.

25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
golf	2	2	1
same sex	1	1	1
out of a	4	2	0.5
said	3	1	0.333333
you are	4	1	0.25
or mental	5	1	0.2
in my	10	2	0.2
rules	15	2	0.133333
black	91	12	0.131868
other than	50	6	0.12
and any	18	2	0.111111
free	111	11	0.0990991
but not	283	28	0.0989399
such as	701	60	0.085592
attempted	13	1	0.0769231
in his	26	2	0.0769231
like	144	11	0.0763889
oriented	27	2	0.0740741
task	14	1	0.0714286
including	733	52	0.0709413
until the	32	2	0.0625
known as	50	3	0.06
was an	17	1	0.0588235
view	68	4	0.0588235
including the	154	9	0.0584416

Tabla 11. Hipónimos: 25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados.

25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:

terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
centimorgan	out of a	total	9454	941	0.5
breakpoint	said	author	253	1021	0.333333
father	said	cal	3895	6921	0.333333
patient	said	speak	95	2449	0.333333
program	said	subsequent	1943	946	0.333333
rna	said	functional	505	1670	0.333333
system	said	compressive	3624	3023	0.333333
mirna	in my	lab	638	12233	0.2
seldom	in my	life	8928	3778	0.2
application	rules	effecti	6163	49658	0.133333
common	rules	begun	2075	11560	0.133333
depict	rules	sirna	446	3158	0.133333
empirical	rules	design	13579	3604	0.133333
general	rules	predict	2348	1879	0.133333
general	rules	pseudoexon	2348	9645	0.133333
nmd	rules	study	4684	492	0.133333
plasmid	rules	yep24	2163	5719	0.133333
positional	rules	distinguish	2084	3960	0.133333
rely	rules	built	11557	6184	0.133333
set	rules	computatio	751	29860	0.133333
set	rules	reflect	751	92	0.133333
similar	rules	sirna	1321	3158	0.133333
subunit-specific	rules	ampa	56660	19735	0.133333
activin	black	bars	20640	28939	0.131868
complete	black	symbol	553	28855	0.131868

Tabla 12. Hipónimos: 25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado.

6.5 Holónimos

Cadenas encontradas: 1211344
Cadenas que relacionan un par de términos ambos en Wordnet: 452756
Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet: 473965
Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío: 284623
Relaciones encontradas: 970

Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240
in	110000
to	63535
a	61275
of the	41447
with	41385
for	38183
is	34758
that	32887
by	31883
in the	26372
from	20888
or	15649
on	15466
an	13380
1	12128
this	11911
to the	10059
and the	7562
for the	6250
of a	5675
in a	5615
3	5553

Tabla 13. Holónimos: Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición.

25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
class	59	2	0.0338983
is in the	31	1	0.0322581
on	6363	163	0.0256168
male	108	2	0.0185185
mass	136	2	0.0147059
define	80	1	0.0125
before the	94	1	0.0106383
part of the	128	1	0.0078125
order	150	1	0.00666667
in this	1345	7	0.00520446
some	775	4	0.00516129
3	780	4	0.00512821
during	1243	6	0.00482703
for	14082	65	0.00461582
syndrome	678	3	0.00442478
after	946	4	0.00422833
development	474	2	0.00421941
to	19377	78	0.00402539
of a	3129	12	0.00383509
for a	527	2	0.00379507
each	1693	6	0.003544
1	2114	7	0.00331126
for this	333	1	0.003003
4	384	1	0.00260417
is	9412	24	0.00254994

Tabla 14. Holónimos: 25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados.

25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:

terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
abundant	class	generegulatory	3312	29213	0.0338983
abundant	class	rna	3312	505	0.0338983
abundant	class	small	3312	2089	0.0338983
abundant	class	tiny	3312	3131	0.0338983
activation	class	1a-pi3k-rac2	32	58992	0.0338983
alternative	class	appear	2233	1315	0.0338983
belong	class	iii	1666	479	0.0338983
belong	class	rna	1666	505	0.0338983
belong	class	rnaseiii	1666	14573	0.0338983
belong	class	sequence	1666	193	0.0338983
brn-3	class	recognize	7963	2338	0.0338983
channel	class	encode	4496	64	0.0338983
commonest	class	mutation	1660	79	0.0338983
compound	class	iks	3614	10229	0.0338983
comprise	class	rna	4598	505	0.0338983
consequence	class	genetic	2482	985	0.0338983
culminate	class	mutation	9096	79	0.0338983
date	class	iiibtubulin	7656	36022	0.0338983
define	class	lesion	1605	4255	0.0338983
disease	class	microsatellite	1708	11508	0.0338983
distinct	class	somatic	438	1176	0.0338983
dppiv-like	class	protein	6214	274	0.0338983
effective	class	iii	4540	479	0.0338983
elements	class	repetitive	4339	4068	0.0338983
encode	class	iii	64	479	0.0338983

Tabla 15. Holónimos: 25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado.

6.6 Merónimos

Cadenas encontradas: 1258276
Cadenas que relacionan un par de términos ambos en Wordnet: 458472
Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet: 487910
Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío: 311894
Relaciones encontradas: 478

Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240
in	110000
to	63535
a	61275
with	41385
for	38183
is	34758
that	32887
by	31883
in the	26372
from	20888
as	18894
at	18738
or	15649
on	15466
an	13380
1	12128
this	11911
to the	10059
2	8365
and the	7562
for the	6250
with the	6143

Tabla 16. Merónimos: Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición.

25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
perform a	6	1	0.166667
i	17	1	0.0588235
via the	33	1	0.030303
or the	213	6	0.028169
define	80	1	0.0125
3	780	7	0.00897436
6	340	3	0.00882353
at	6174	51	0.00826045
including	733	5	0.00682128
during the	302	2	0.00662252
10	461	3	0.00650759
is the	676	4	0.00591716
differences in	179	1	0.00558659
under the	182	1	0.00549451
2	1246	6	0.00481541
and a	764	3	0.0039267
or	4902	18	0.00367197
on a	274	1	0.00364964
under	613	2	0.00326264
such as	701	2	0.00285307
when	712	2	0.00280899
one	1608	4	0.00248756
for each	418	1	0.00239234
and	38072	90	0.00236394
into	1391	3	0.00215672

Tabla 17. Merónimos: 25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados.

25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:

terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
apc	perform a	similar	17975	1321	0.166667
author	perform a	careful	1021	2665	0.166667
family	perform a	genetic	775	985	0.166667
gcyto-actin	perform a	critical	27793	1634	0.166667
opportunity	perform a	prenatal	12487	3893	0.166667
pcr	perform a	padlock	148	4269	0.166667
rvation	perform a	bioinformatic	8226	2139	0.166667
sequence	perform a	tblastn	193	8848	0.166667
servation	perform a	bioinformatic	10172	2139	0.166667
study	perform a	systematic	492	2318	0.166667
suggestion	perform a	prenatal	4297	3893	0.166667
uence	perform a	pulldown	1731	2545	0.166667
cell	i	p27kip1	360	13437	0.0588235
creu	i	sant	57691	57692	0.0588235
deletion	i	trkc	202	18053	0.0588235
duran	i	reynal	57688	57689	0.0588235
egf	i	inesin	71	57697	0.0588235
egf	i	mcf	71	57696	0.0588235
gens	i	malaltia	57694	57695	0.0588235
haplotype	i	population	354	419	0.0588235
hybridization	i	situ	5624	1071	0.0588235
indice	i	represent	6160	561	0.0588235
information	i	cyclic	2881	4855	0.0588235
lower	i	250dystrophin	623	57693	0.0588235
manipulation	i	vitro	11671	1671	0.0588235

Tabla 18. Merónimos: 25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado.

6.7 Dominios

Cadenas encontradas: 1311812
Cadenas que relacionan un par de términos ambos en Wordnet: 486654
Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet: 516039
Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío: 309119
Relaciones encontradas: 29

Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240
in	110000
to	63535
a	61275
of the	41447
with	41385
for	38183
is	34758
that	32887
by	31883
in the	26372
are	21002
from	20888
as	18894
at	18738
or	15649
on	15466
an	13380
1	12128
this	11911
have	10447
to the	10059
2	8365

Tabla 19. Dominios: Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición.

25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
of an	553	1	0.00180832
and	38072	15	0.00039399
are	6472	2	0.000309023
an	3888	1	0.000257202
in	46558	5	0.000107393
of the	19170	1	0.0000521648
of	78848	3	0.0000380479
the	76680	1	0.0000130412
a cornerstone of american legal	0	0	0
zone between	0	0	0
a	22963	0	0
a batted or thrown	0	0	0
a broadly encompassing	0	0	0
a battle in	0	0	0
a big crowd	0	0	0
2005 is an	0	0	0
1410 rise of	0	0	0
1809 vienna	0	0	0
1916 ends in	0	0	0
1917 in	0	0	0
1967 williams	0	0	0
1991	2	0	0
2	1246	0	0
2000	29	0	0
2005	65	0	0

Tabla 20. Dominios: 25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados.

25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:

terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
30-utr	of an	mrna	3879	887	0.00180832
30overhang	of an	sirna	13708	3158	0.00180832
ability	of an	hspka	1340	12697	0.00180832
absence	of an	abp	4317	4875	0.00180832
absence	of an	exogenou	4317	12344	0.00180832
absence	of an	inhibitory	4317	12388	0.00180832
absence	of an	ish	4317	10835	0.00180832
absence	of an	sc35	4317	39	0.00180832
accumulation	of an	fusi	2711	13105	0.00180832
acetylation	of an	acetylase	2487	13110	0.00180832
acquisition	of an	eyeblink	2538	8432	0.00180832
action	of an	inhibitory	2082	12388	0.00180832
activation	of an	additional	32	460	0.00180832
activation	of an	xbp1dependent	32	12987	0.00180832
activity	of an	antisense	1822	156	0.00180832
activity	of an	intracellular	1822	1562	0.00180832
activity	of an	mirna	1822	638	0.00180832
adation	of an	increase	8606	3359	0.00180832
addition	of an	ecori	1379	8847	0.00180832
addition	of an	oligomer	1379	2816	0.00180832
addition	of an	sequ	1379	12365	0.00180832
age	of an	artificial	473	12369	0.00180832
allele	of an	genetic	480	985	0.00180832
allele	of an	mmr	480	6881	0.00180832
amdepend	of an	agonist	13262	13263	0.00180832

Tabla 21. Dominios: 25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado.

6.8 Miembros

Cadenas encontradas: 1341947
Cadenas que relacionan un par de términos ambos en Wordnet: 496463
Cadenas que relacionan un par de términos en los que al menos uno no se encuentra en Wordnet: 529576
Cadenas que relacionan un par de términos en los que al menos uno de los términos es vacío: 315908
Relaciones encontradas: 42

Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición:

cadena_intermedia	apariciones_totales
the	196392
of	175568
and	132240
in	110000
to	63535
a	61275
of the	41447
with	41385
for	38183
is	34758
that	32887
by	31883
in the	26372
was	24240
from	20888
as	18894
at	18738
or	15649
on	15466
an	13380
not	12186
this	11911
to the	10059
2	8365
and the	7562

Tabla 22. Miembros: Primeras 25 cadenas intermedias encontradas en total ordenadas en orden decreciente de aparición.

25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados:

cadena_intermedia	num_apariciones	num_relaciones	fiabilidad
if a	18	2	0.111111
on a	274	1	0.00364964
2	1246	1	0.000802568
and the	3170	2	0.000630915
at the	1826	1	0.000547645
and	38072	11	0.000288926
to the	4200	1	0.000238095
a	22963	4	0.000174193
at	6174	1	0.00016197
on	6363	1	0.000157159
the	76680	8	0.00010433
to	19377	2	0.000103215
for	14082	1	0.0000710126
of	78848	4	0.0000507305
in	46558	2	0.0000429572
a member of the	12	0	0
a lateral	3	0	0
a misplay such as a	0	0	0
a plant or a	0	0	0
8 lb cast iron	0	0	0
4 3 in the second	0	0	0
2k7	0	0	0
your	50	0	0
2 player drills	0	0	0
16 deg v foil m455	0	0	0

Tabla 23. Miembros: 25 primeras cadenas ordenadas en orden decreciente de fiabilidad que tenían términos de Wordnet a ambos lados.

25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado:

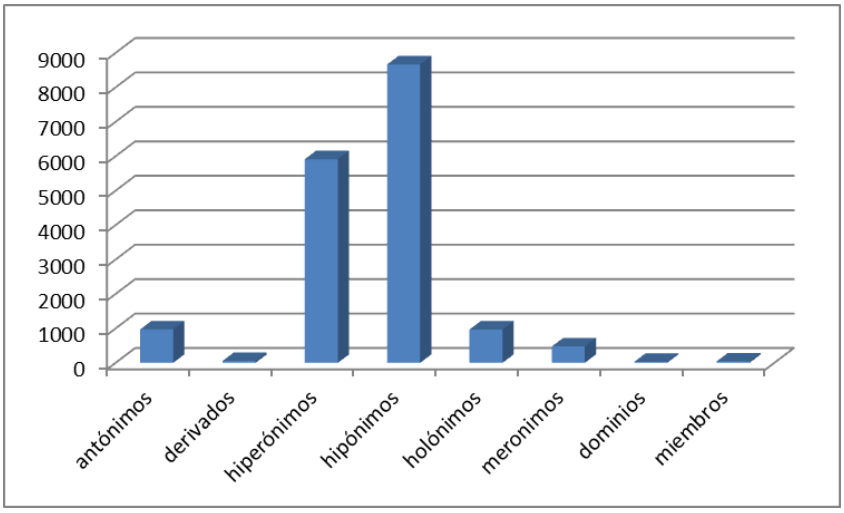
terminoA	cadena_intermedia	terminoB	id_hijo	id_padre	fiabilidad
absent	if a	kinase-dead	1597	10332	0.111111
behave	if a	critical	2825	1634	0.111111
bind	if a	conformational	4864	630	0.111111
determine	if a	fetus	29	7118	0.111111
determine	if a	kinetin	29	85	0.111111
determine	if a	locus	29	2515	0.111111
determine	if a	sample	29	434	0.111111
determine	if a	small	29	2089	0.111111
identify	if a	hit	1140	2147	0.111111
logic	if a	vestibular	9338	5354	0.111111
member	if a	kv1	394	5520	0.111111
straightforward	if a	mutation	3730	79	0.111111
suggest	if a	large	1017	1981	0.111111
unclear	if a	loss	4350	486	0.111111
worm	if a	functional	3192	1670	0.111111
worth	if a	mirna	8601	638	0.111111
17q11	on a	pcr	7443	148	0.00364964
17q11	on a	pcr-base	7443	7445	0.00364964
308c	on a	platform	6758	3524	0.00364964
48c	on a	wheel	3278	8332	0.00364964
acetate	on a	jeol	3412	2760	0.00364964
addition	on a	trypan	1379	6071	0.00364964
advantage	on a	abundant	4923	3312	0.00364964
advantage	on a	computational	4923	590	0.00364964
al68	on a	patient	7488	95	0.00364964

Tabla 24. Miembros: 25 primeros pares de términos en los que al menos uno de ellos no se encuentra en Wordnet, la cadena intermedia que los relaciona, sus identificadores asignados como hijo y padre y la fiabilidad de establecer relación del tipo estudiado.

6.9 Miscelánea

En este apartado se muestran algunos datos que no se han recogido en los apartados anteriores y que nos parecen interesantes.

En la siguiente gráfica podemos ver una comparación con el número de relaciones encontradas de cada tipo de relación:



Se puede comprobar que las relaciones opuestas de hiperonimia e hiponimia son las que más pares de términos encuentran. Esto tiene sentido atendiendo por un lado al número de entradas que tenía cada tabla de la base de datos de Wordnet y por otro lado a la naturaleza de la relación estudiada. En cuanto al primer aspecto, el número de entradas para cada tipo de relación que se encontraban en la base de datos de Wordnet era:

Relación de Antónimos: 1940

Relación de Derivados: 2695

Relación de Hiperónimos: 83646

Relación de Hipónimos: 83649

Relación de Holónimos: 21929

Relación de Merónimos: 21929

Relación de Dominios: 6251

Relación de Miembros: 6252

Como se puede comprobar, las relaciones de hiperonimia e hiponimia son las que más pares de palabras tenían registrados en base de datos y son las que más han sido encontradas en los textos sobre la sordera genética.

A continuación una tabla de algunos de los términos que no se han encontrado en la base de datos de Wordnet y que podrían ser candidatos a formar parte de un tesauro especializado sobre la deficiencia auditiva:

identificador	término
2	electrophoretic
22	transglutaminase
24	myotubularin
33	cryptic
40	polymorphic
59	hexamer
63	somatostatin
68	neocortical
89	transfection
138	pseudogene
240	lamellar
270	desmoglein
284	polyubiquitin
290	tubulin
349	na-pyruvate
354	haplotype
357	pyruvate
365	homozygote
381	presenilin
431	calsequestrin
468	transgenic
501	hyperlocomotion
540	demethylation
595	mitofusin

Si buscamos la definición de algunos de estos términos en la web podemos encontrar (directamente del inglés):

Electrophoretic: the motion of dispersed particles relative to a fluid under the influence of a spatially uniform[4].

Transglutaminase: enzyme that catalyzes the formation of an isopeptide bond between a free amine group (e.g., protein- or peptide-bound lysine) and the acyl group at the end of the side chain of protein- or peptide-bound glutamine[5].

myotubularin : myotubularin domain represents a region within eukaryotic myotubularin-related proteins that is sometimes found with the GRAM domain[6].

hexamer: molecule made up of six structural subunits[7].

somatostatin : also known as growth hormone–inhibiting hormone (GHIH) or by several other names, is a peptide hormone that regulates the endocrine system and affects neurotransmission and cell proliferation via interaction with G protein-coupled somatostatin receptors and inhibition of the release of numerous secondary hormones. Somatostatin inhibits insulin and glucagon secretion[8].

transfection: is the process of deliberately introducing nucleic acids into cells[9].

pseudogene: functionless relatives of genes that have lost their gene expression in the cell or their ability to code protein[10].

---

[4] Electrophoresis. (2016, January 15). In Wikipedia, The Free Encyclopedia. Retrieved 17:32, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Electrophoresis&oldid=699937637

[5] Transglutaminase. (2016, January 11). In Wikipedia, The Free Encyclopedia. Retrieved 17:34, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Transglutaminase&oldid=699254746

[6] Myotubularin. (2014, February 11). In Wikipedia, The Free Encyclopedia. Retrieved 17:35, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Myotubularin&oldid=594963955

[7] Random hexamer. (2014, March 17). In Wikipedia, The Free Encyclopedia. Retrieved 17:37, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Random_hexamer&oldid=600052072

[8] Somatostatin. (2016, February 28). In Wikipedia, The Free Encyclopedia. Retrieved 17:38, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Somatostatin&oldid=707357806

[9] Transfection. (2016, February 22). In Wikipedia, The Free Encyclopedia. Retrieved 17:39, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Transfection&oldid=706195970

[10] Pseudogene. (2016, February 8). In Wikipedia, The Free Encyclopedia. Retrieved 17:39, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Pseudogene&oldid=703967557

lamellar : lamellar structures or microstructures are composed of fine, alternating layers of different materials in the form of lamellae[11].

desmoglein: family of cadherins consisting of proteins DSG1, DSG2, DSG3, and DSG4[12].

tubulin: in molecular biology can refer either to the tubulin protein superfamily of globular proteins, or one of the member proteins of that superfamily. á- and â-tubulins polymerize into microtubules, a major component of the eukaryotic cytoskeleton[13].

Na-pyruvate: Sodium pyruvate is commonly added to cell culture media as an additional source of energy, but may also have protective effects against hydrogen peroxide[14].

Haplotype: in the simplest terms, a specific group of genes or alleles that progeny inherited from one parent[15].

Pyruvate: Pyruvic acid (CH3COCOOH) is the simplest of the alpha-keto acids, with a carboxylic acid and a ketone functional group. Pyruvate, the conjugate base, CH3COCOO−, is a key intermediate in several metabolic pathways[16].

Presenilin: Presenilins are a family of related multi-pass transmembrane proteins which constitute the catalytic subunits of the gamma-secretase intramembrane protease complex[17].

Calsequestrin: calcium-binding protein of the sarcoplasmic reticulum[18].

---

[11] Lamellar structure. (2013, May 11). In Wikipedia, The Free Encyclopedia. Retrieved 17:39, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Lamellar_structure&oldid=554631021

[12] Desmoglein. (2015, August 31). In Wikipedia, The Free Encyclopedia. Retrieved 17:40, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Desmoglein&oldid=678775414

[13] Tubulin. (2016, January 29). In Wikipedia, The Free Encyclopedia. Retrieved 17:40, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Tubulin&oldid=702188629

[14] Sodium pyruvate. (2013, August 25). In Wikipedia, The Free Encyclopedia. Retrieved 17:41, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Sodium_pyruvate&oldid=570181277

[15] Haplotype. (2016, January 24). In Wikipedia, The Free Encyclopedia. Retrieved 17:41, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Haplotype&oldid=701378216

[16] Pyruvic acid. (2016, February 7). In Wikipedia, The Free Encyclopedia. Retrieved 17:43, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Pyruvic_acid&oldid=703799488

[17] Presenilin. (2015, October 8). In Wikipedia, The Free Encyclopedia. Retrieved 17:44, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Presenilin&oldid=684674787

[18] Calsequestrin. (2014, January 15). In Wikipedia, The Free Encyclopedia. Retrieved 17:45, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Calsequestrin&oldid=590848083

Demethylation: the chemical process resulting in the removal of a methyl group (CH3) from a molecule.[1][2] A common way of demethylation is the replacement of a methyl group by a hydrogen atom, resulting in a net loss of one carbon and two hydrogen atoms[19].

Mitofusin(MFN2): Mitofusin-2 is a protein that in humans is encoded by the MFN2 gene[20].

[19] Demethylation. (2015, September 25). In Wikipedia, The Free Encyclopedia. Retrieved 17:45, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=Demethylation&oldid=682763752

[20] MFN2. (2014, January 26). In Wikipedia, The Free Encyclopedia. Retrieved 17:46, March 4, 2016, from https://en.wikipedia.org/w/index.php?title=MFN2&oldid=592483024

7. CONCLUSIONES

7.1 Conclusiones

En un mundo como el actual en el que las relaciones entre máquinas y humanos son cada vez más numerosas y complejas y en el que las propias máquinas han de compartir cada vez más información entre sí, se hace necesario dotar a estas de mecanismos que las permitan ir más allá del mero tratamiento de datos y haciéndolas capaces de interpretar, de alguna manera, los conceptos que hay detrás de ellos y poder así acercarse más a la forma en que se comunican las personas.

Es una tarea ardua y compleja conseguir que una máquina entienda qué quiere decir una persona, qué está intentando buscar, dónde quiere llegar a parar, máxime cuando la propia comunicación entre humanos está llena de ambigüedades y formas subjetivas de interpretar las cosas.

El proceso de creación de una web semántica está aún en desarrollo. Las ontologías y tesauros son herramientas que ayudan a avanzar en este objetivo al presentar el conocimiento humano de una manera organizada. Los vocabularios presentados en cada una de ellas así como las relaciones que se dan entre sus miembros, hacen que se puedan usar como base sobre la que ir construyendo herramientas que faciliten el manejo de conceptos semánticos por las máquinas.

En este trabajo se plantearon dos objetivos:

1. Para cada tipo de relación entre palabras estudiado (Antónimos, Derivados, Hiperónimos, Hipónimos, Holónimos, Merónimos, pertenencia a un dominio y miembros de un dominio) comprobar la capacidad de una seria de cadenas de texto de establecer un tipo concreto de esas ocho relaciones.

   Esto se ha conseguido y representado en el capítulo 6, *Resultados*, en las dos primeras tablas que aparecen por cada tipo de relación, donde se pueden ver las primeras entradas recogidas de la base de datos creada en este proyecto.

2. El segundo objetivo que se planteaba era encontrar términos no recogidos en Wordnet y establecer una probabilidad de establecer uno de los ocho tipos de relación estudiados con otra palabra que podría estar a su vez en Wordnet o no.

   En la tercera tabla, para cada tipo de relación, del capítulo 6, *Resultados,* se encuentra una pequeña muestra de la información obtenida y almacenada en la base de datos creada en este proyecto.

7.2 Actuaciones futuras

Habiendo finalizado este proyecto podemos sugerir algunas líneas de actuación o mejoras futuras tras el mismo:

- A nivel de software:

    - Se podría incorporar un analizador sintáctico en la parte que extrae los términos a la derecha e izquierda de las cadenas intermedias para encontrar sustantivos con una mayor exactitud.

    - Se podrían extraer frases más largas a analizar para evitar posibles palabras cortadas o la no obtención de términos a izquierda o derecha de las cadenas estudiadas.

    - Se podría intentar pasar a plural tanto las cadenas intermedias como las frases extraídas a su derecha e izquierda y volver a pasar todo a singular una vez extraídos los términos a izquierda y derecha de las cadenas intermedias.

- A nivel teórico:

    - Se podrían contrastar con un experto en la materia de la sordera genética los resultados obtenidos.
    - Llegados a cierto punto, elaborar formalmente un tesauro sobre el campo de conocimiento de la sordera genética.

8. BIBLIOGRAFIA

Web Semántica

http://www.puromarketing.com/12/15656/social-semantica.html

https://es.wikipedia.org/wiki/Web_sem%C3%A1ntica

http://www.humanlevel.com/articulos/desarrollo-web/el-futuro-de-la-websemantica.html

Tesauros

http://www.ricardotayar.com

http://pendientedemigracion.ucm.es/info/multidoc/prof/fvalle/tesauro.htm

http://www.apuntes.eu/otras-materias/estructura-del-tesauro-las-relaciones-semanticas/

Ontologías

http://www.hipertexto.info/documentos/ontologias.htm

http://www.infor.uva.es/~sblanco/Tesis/Ontolog%C3%ADas.pdf

https://sites.google.com/site/jojooa/inteligencia-artificial/componentes-de-una-ontologia

http://sedici.unlp.edu.ar/bitstream/handle/10915/23076/Documento_completo.pdf?sequence=1

Wordnet

https://wordnet.princeton.edu/

http://elies.rediris.es

Relaciones semánticas

https://es.wikipedia.org/wiki/Relaci%C3%B3n_sem%C3%A1ntica

http://www.ecured.cu/Relaci%C3%B3n_sem%C3%A1ntica

JAVA

http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html

http://www.webtaller.com/manual-java/caracteristicas-java.php

ECLIPSE

http://www.genbetadev.com/herramientas/eclipse-ide

http://es.slideshare.net/MagaLasic/presentacion-eclipse-grupo-6

MySQL

http://culturacion.com/que-es-mysql/

MySQL Workbench

http://gizmos.republica.com/programas-y-aplicaciones/mysql-workbench-editor-visual-de-bases-de-datos-mysql.html

http://www.tecnopedia.net/mysql/mysql-workbench-5/

Resultados: definiciones

http://www.wikipedia.es